

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



Conector OutSystems para
Oracle E-Business Suite

Ivan David de Almeida Soares

Mestrado em Engenharia Informática

2008

UNIVERSIDADE DE LISBOA

Faculdade de Ciências
Departamento de Informática



Conector OutSystems para
Oracle E-Business Suite

Ivan David de Almeida Soares

PROJECTO

Projecto orientado pelo Prof^ª. Dr^a. Teresa Chambel
e co-orientado pelo Dr. João Campos

Mestrado em Engenharia Informática

2008

Declaração

Ivan Soares, aluno nº 26538 da Faculdade de Ciências da Universidade de Lisboa, declara ceder os seus direitos de cópia sobre o seu Relatório de Projecto em Engenharia Informática, intitulado "Conector OutSystems para Oracle E-Business Suite", realizado no ano lectivo de 2007/2008 à Faculdade de Ciências da Universidade de Lisboa para o efeito de arquivo e consulta nas suas bibliotecas e publicação do mesmo em formato electrónico na Internet.

FCUL, 30 de Setembro de 2008

João Campos, supervisor do projecto de Ivan Soares, aluno da Faculdade de Ciências da Universidade de Lisboa, declara concordar com a divulgação do Relatório do Projecto em Engenharia Informática, intitulado "Conector OutSystems para Oracle E-Business Suite".

Lisboa, 30 de Setembro de 2008

Resumo

Actualmente, verifica-se no mercado, um investimento crescente por parte das organizações na aquisição e implementação de sistemas aplicacionais integrados. Pretendem com esse investimento simplificar e otimizar a gestão dos processos associados à sua actividade. Da oferta existente, muitas destas organizações optam pelo sistema aplicacional Oracle E-Business Suite. O dinamismo imposto pelo mercado e o facto deste ser um sistema genérico e abrangente, torna necessário desenvolver novas aplicações periféricas em redor deste sistema nuclear, tarefa dificultada pela necessidade de especialização técnica que é por inerência, dispendiosa.

Este projecto endereça esta problemática, tendo como objectivo dotar a plataforma de desenvolvimento ágil OutSystems de uma camada capaz de possibilitar a utilização directa da lógica e entidades de negócio do Oracle E-Business Suite directamente no ambiente de desenvolvimento, capacitando a produção de novas extensões e outras aplicações periféricas, acelerando o seu desenvolvimento e reduzindo custos.

O protótipo resultante deste projecto foi testado e aplicado na criação de uma nova extensão ao Oracle E-Business Suite que, empregando a metodologia Scrum, foi desenvolvido com custos muito inferiores aos previstos num desenvolvimento clássico.

Este projecto foi realizado no âmbito da disciplina de Projecto em Engenharia Informática do Mestrado em Engenharia Informática da Faculdade de Ciências da Universidade de Lisboa e desenvolvido na PRÁXIA, Sistemas de Informação S.A..

PALAVRAS-CHAVE: OutSystems, Oracle E-Business Suite, Integração, Conector, Aplicações Periféricas

Abstract

There is a growing investment by organizations on integrated core financial and resource planning systems. These systems drive key company management decisions on how to improve their operation.

The available commercial solutions for these systems include Oracle E-Business Suite. However, this is a generic system, able to support food manufacturing as well as airplane maintenance. It must be configured to fulfill company requirements and only with extended customization, can this platform adapt to company internal procedures and operation. The problem is that, extended customization of the Oracle E-Business Suite requires specialized teams that have limited productivity and are expensive.

The OutSystems Connector addresses this problem. It provides a integration layer between the OutSystems agile development platform and Oracle E-Business Suite. The integration layer provides E-Business Suite actions and functionalities directly to developers. With this integration layer, common developers are not aware of any technical details needed to implement complex business functions on the E-Business Suite. They simply use them to fulfill new extension and customization requirements to core systems reducing time and the resources needed to develop new applications. The OutSystems Connector uses Oracle E-Business Suite knowledge as well as programmatic interfaces and conversion logic to accomplish this.

The prototype that was built, was tested and used to create new applications on the Oracle E-Business Suite. Using Scrum development methods and the Connector, the application was built in less time and with less resources than other estimates using Oracle E-Business Suite traditional techniques.

This document describes the project implemented for the course Project in Informatics Engineering of the Masters in Informatics Engineering of the Faculty of Sciences at Lisbon University.

KEYWORDS: OutSystems, Oracle E-Business Suite, Integration, Connector, External Applications

Conteúdo

Conteúdo	xi
Lista de Figuras	xiii
Lista de Tabelas	xv
Capítulo 1 Introdução	1
1.1 Motivação	1
1.2 Objectivos	2
1.3 Resumo do Trabalho Desenvolvido	3
1.4 Enquadramento Institucional	3
1.5 Estrutura do Relatório	4
Capítulo 2 Tecnologia e Trabalho Relacionado	5
2.1 Sistemas Aplicacionais Integrados	5
2.1.1 <i>ERP - Enterprise Resource Planning</i>	5
2.1.2 <i>CRM - Customer Relationship Management</i>	6
2.1.3 <i>SCM - Supply-Chain Management</i>	7
2.1.4 <i>PLM - Product Lifecycle Management</i>	7
2.1.5 <i>SRM - Supplier Relationship Management</i>	8
2.2 Soluções Aplicacionais Integradas Existentes	8
2.2.1 Oracle E-Business Suite	9
2.2.2 SAP Business Suite	11
2.2.3 Outras Soluções Aplicacionais Integradas	11

2.2.4	Mercado dos Sistemas Aplicacionais Integrados.....	12
2.3	Plataforma de Desenvolvimento OutSystems.....	12
2.3.1	Plataforma OutSystems.....	12
2.3.2	Agile Solutions for SAP.....	17
Capítulo 3	Metodologia e Planeamento.....	21
3.1	Metodologia de Desenvolvimento em Espiral.....	21
3.2	Metodologia de Desenvolvimento Scrum.....	22
3.2.1	Papéis	24
3.2.2	<i>Sprinting</i>	25
3.2.3	Artefactos	27
3.2.4	Discussão	28
3.3	Planeamento.....	30
Capítulo 4	Arquitectura do Sistema.....	35
4.1	Arquitectura Global	35
4.1.1	Sistemas Base.....	36
4.1.2	Plataforma OutSystems.....	38
4.1.3	Generator.....	39
4.1.4	OEBS Transparency Services	41
4.1.5	Composite Applications.....	42
4.2	Modelo de Segurança.....	42
Capítulo 5	Trabalho Realizado e Resultados.....	45
5.1	Integração e Configuração	45
5.2	Connector e Aplicação Generator.....	46
5.3	Aplicação Receivables Workbench	56
5.4	Resultados Alcançados	61
Capítulo 6	Conclusão e Trabalho Futuro.....	65
Bibliografia		69

Lista de Figuras

Figura 1 - Arquitectura OutSystems	15
Figura 2 - Processo de <i>Deploy (eSpace/Extension)</i>	16
Figura 3 - Transparency Services for SAP	19
Figura 4 - Metodologia de desenvolvimento em Espiral	22
Figura 5 - Metodologia de desenvolvimento Scrum.....	28
Figura 6 - Planeamento inicial	30
Figura 7 - Planeamento final.....	30
Figura 8 - <i>Sprint Backlog</i>	33
Figura 9 - Análise do <i>Sprint Backlog</i>	34
Figura 10 - Arquitectura Global do Sistema	36
Figura 11 - Arquitectura Geral OEBS	37
Figura 12 - Arquitectura da Instância da Plataforma OutSystems do Projecto	38
Figura 13 - Representação gráfica do componente Generator - conexão entre a plataforma OutSystems e o OEBS	40
Figura 14 - Principais funções do OEBS Transparency Services.....	41
Figura 15 - Composite Application - Receivables Workbench	42
Figura 16 - Modelo de Segurança para limitar o acesso da plataforma OutSystems ao OEBS	44
Figura 17 - Processo Gerador	48
Figura 18 - Generator - Pesquisar e Adicionar	50

Figura 19 - Generator - Remove	50
Figura 20 - Generator - Gerar	51
Figura 21 - Generator - <i>Download</i>	51
Figura 22 - Generator - Ficheiro gerados	52
Figura 23 - Receivables Workbench - Modelo de Dados	57
Figura 24 - Receivables Workbench - Gestão de Recebimentos	58
Figura 25 - Receivables Workbench - Criar Recebimento	59
Figura 26 - Receivables Workbench - Classificar Transacção	59
Figura 27 - Ambiente de desenvolvimento Service Studio - Fluxo da <i>Action</i> Criar_Recebimento	60
Figura 28 - Ambiente de desenvolvimento Service Studio - Interface do utilizador para a gestão de recebimentos	61

Lista de Tabelas

Tabela 1 - Planeamento final detalhado	32
Tabela 2 - Tipos de Dados	54
Tabela 3 - Receivables Workbench - Análise do tempo estimado e efectivo de desenvolvimento	62

Capítulo 1

Introdução

Este documento apresenta o projecto do Conector OutSystems para Oracle E-Business Suite que permite a ligação entre a plataforma de desenvolvimento ágil OutSystems e o sistema aplicacional Oracle E-Business Suite, no âmbito do Mestrado em Engenharia Informática da Faculdade de Ciências da Universidade de Lisboa, desenvolvido na PRÁXIA, Sistemas de Informação S.A..

1.1 Motivação

As empresas e organizações sentem actualmente uma forte necessidade e dependência de sistemas de gestão aplicacionais integrados para gerirem e optimizarem os processos de negócio associados às suas actividades, automatizando tarefas repetitivas e libertando recursos para actividades geradoras de valor ao negócio.

A Oracle é uma das companhias que comercializa este tipo de sistemas, disponibilizando um conjunto de aplicações, o Oracle E-Business Suite, com vista a simplificar, automatizar e dinamizar os processos de negócio. Estas aplicações são genéricas, abrangendo a maior parte das funcionalidades necessárias a grande parte das organizações, independentemente do sector, área de negócio e estrutura, sendo mesmo independentes de algumas condicionantes culturais como a língua falada pelos utilizadores.

Estes sistemas genéricos incluem um leque amplo de funções e opções, levando a que as organizações sintam a necessidade de personalizar as aplicações disponíveis e criar

novas aplicações periféricas aos sistemas nucleares, que implementam funções específicas do negócio, à medida das suas necessidades e adaptadas ao dinamismo imposto pelo mercado. Contudo, a criação de novas aplicações para integrar e interagir com estes sistemas torna-se difícil, pois requer uma especialização técnica e conhecimento profundo da estrutura funcional destas aplicações, requisitos que não são compatíveis com as necessidades de desenvolvimento rápido e a custos reduzidos.

Actualmente, para o Oracle E-Business Suite, não se encontram disponíveis ferramentas que permitam criar e desenvolver aplicações periféricas de forma rápida, ágil e com custos reduzidos. O Conector OutSystems para Oracle E-Business Suite tenta responder a estes requisitos. Este projecto, tira partido da tecnologia OutSystems, uma plataforma tecnológica assente em metodologias de desenvolvimento ágil que oferece flexibilidade, rapidez e simplicidade de desenvolvimento, para criar uma ferramenta de integração com o Oracle E-Business Suite que permitirá desenvolver aplicações periféricas, de forma rápida e com custos reduzidos, encapsulando a complexidade natural deste sistema.

1.2 Objectivos

O objectivo principal deste projecto consiste em desenvolver um conector de integração do sistema aplicacional Oracle E-Business Suite, com a plataforma OutSystems, facilitando o desenvolvimento de futuras aplicações *Web* em tecnologia OutSystems integradas com o sistema nuclear Oracle E-Business Suite.

Este conector permitirá criar aplicações *Web* que poderão explorar toda a estrutura de base de dados onde está centralizada toda a informação do Oracle E-Business Suite, bem como a lógica de negócio nuclear das aplicações, reduzindo a complexidade e minimizando as necessidades específicas de integração com aplicações periféricas. Com este conector será possível a construção de aplicações de forma ágil e célere, permitindo alcançar os requisitos de rapidez e custo reduzido impostos pelos objectivos de minimização de custos e maximização de resultados que obrigam as organizações a serem flexíveis e capazes de abraçar a mudança para sobreviver.

1.3 Resumo do Trabalho Desenvolvido

Neste projecto, além do componente de *software* de integração (Connector) da plataforma OutSystems com o Oracle E-Business Suite, foram também desenvolvidas duas aplicações *Web* através da tecnologia OutSystems.

O Connector é composto por um conjunto de métodos OutSystems que interagem, de uma forma transparente, com os procedimentos ou funções existentes na base de dados Oracle, contendo toda a lógica de negócio do Oracle E-Business Suite.

A primeira aplicação desenvolvida, Generator, é composta por uma interface intuitiva e um conjunto de métodos desenvolvidos na linguagem C# .NET, responsáveis por gerar de uma forma automática o conjunto de métodos que compõem o Connector.

Foi ainda desenvolvida uma segunda aplicação, denominada Receivables Workbench, para automação e simplificação de procedimentos relacionados com a gestão de tesouraria e recebimentos, que usa o Connector gerado para o acesso à lógica de negócio do Oracle E-Business Suite. Esta aplicação permite efectuar operações funcionais financeiras, nomeadamente registar recebimentos e efectuar transacções de receita, mantendo toda a informação gerada disponível aos diversos módulos do Oracle E-Business Suite.

1.4 Enquadramento Institucional

Este projecto iniciou-se em Setembro de 2007 com uma fase de integração na empresa PRÁXIA, Sistemas de Informação, S.A., tendo como objectivos dar a conhecer a estrutura da empresa, as competências de cada unidade orgânica, bem como os seus métodos de trabalho, as normas internas, a política de qualidade e toda a documentação a ser produzida e utilizada no desenvolvimento de projectos.

A PRÁXIA é uma empresa de Sistemas de Informação com fortes competências nas áreas de desenvolvimento de *software*, formação, administração e suporte de sistemas de informação, planeamento, gestão, acompanhamento e validação da qualidade de projectos de informática.

O projecto foi desenvolvido nas instalações de um dos clientes da PRÁXIA, onde o autor integrou uma equipa que presta serviços sobre Oracle E-Business Suite,

permitindo a exploração e contextualização necessárias à sua concretização. A instituição em causa tem implementada uma solução de gestão de negócio baseada no Oracle E-Business Suite, utilizando para a gestão dos seus processos financeiros o módulo Financials.

1.5 Estrutura do Relatório

Após uma breve introdução ao projecto, referindo os motivos que estiveram na base da escolha do tema e da solução apresentada, segue-se o Capítulo 2 onde é efectuado um enquadramento tecnológico referindo trabalhos relacionados, as tecnologias envolvidas, e uma breve análise de soluções similares. No Capítulo 3 é apresentada a metodologia utilizada no desenvolvimento do projecto e é efectuada a confrontação do planeamento original com o que realmente foi executado, seguindo-se a apresentação da arquitectura da solução no Capítulo 4. No Capítulo 5 é detalhado o trabalho realizado, terminando com uma análise dos resultados obtidos. Para finalizar é efectuada uma conclusão e identificadas possíveis direcções para trabalhos futuros.

Capítulo 2

Tecnologia e Trabalho Relacionado

Neste capítulo apresentam-se os sistemas e as tecnologias envolvidas no desenvolvimento do Conector OutSystems para Oracle E-Business Suite.

2.1 Sistemas Aplicacionais Integrados

Sistemas Aplicacionais Integrados são conjuntos de aplicações integradas destinadas à gestão de negócio dos mais variadíssimos sectores empresariais. São descritas de seguida as principais famílias destes sistemas.

2.1.1 *ERP - Enterprise Resource Planning*

Enterprise Resource Planning (Planeamento de Recursos Empresariais) [7] são sistemas de informação transaccionais (*OLTP - Online Transaction Processing*) integrados que centralizam todos os dados e processos de uma organização num único sistema, cuja função é armazenar, processar e organizar a informação gerada pelos processos de negócio, agregando e estabelecendo relações de informação entre todas as áreas de uma organização.

Os sistemas *ERP* são compostos por uma base de dados única e por módulos integrados que suportam as diversas actividades das organizações. Estes módulos armazenam toda a informação na base de dados central permitindo a partilha da mesma com os restantes módulos existentes.

O acesso à informação centralizada, assim como a sua actualização, podem ser efectuados em qualquer instante, através de qualquer um dos módulos integrados do sistema, possibilitando desta forma acelerar os processos de negócio, partilhar toda informação pela organização sempre que necessário e evitando a redundância de processos.

Os principais módulos de aplicações que constituem este tipo de sistemas são: manufacturação; financeiro; recursos humanos; gestão da cadeia de fornecimento; projectos e gestão da relação com o cliente.

Aos sistemas *ERP* podem ser associadas ferramentas de *Business Intelligence* [3] com o objectivo de transformar os dados gerados e armazenados, em informação de gestão, recolhendo indicadores e medidas de desempenho, possibilitando a realização de planos de acção e melhorias.

A implementação deste tipo de sistemas tem como principais benefícios a redução de custos, a optimização do fluxo da informação e a qualidade da mesma, a optimização do processo de tomada de decisão, a eliminação da redundância de actividades, a capacidade de partilha de informação por qualquer departamento da organização, assim como a redução dos prazos de resposta ao mercado.

2.1.2 CRM - Customer Relationship Management

Customer Relationship Management (Gestão do Relacionamento com o Cliente) [6] são sistemas constituídos por um conjunto de procedimentos/processos organizados e integrados num modelo de gestão de negócio. Estes sistemas são destinados à gestão e automatização de todos os processos de relacionamento com o cliente.

Os sistemas *CRM* permitem automatizar e tornar mais eficientes as áreas de negócio associadas ao marketing, produtos, vendas, serviço e apoio ao cliente, melhorando também o rendimento e a eficácia dos processos da organização, disponibilizando informação fiável e precisa sobre o cliente.

Os quatro grandes objectivos/benefícios da implementação de uma solução *CRM* são: a obtenção de novos clientes; a fidelização dos clientes existentes; a análise contínua da rentabilidade de cada cliente; a redução dos custos através da definição de objectivos específicos para satisfazer as necessidades individuais de cada cliente por intermédio da oferta dos produtos e serviços mais adequados.

2.1.3 *SCM - Supply-Chain Management*

Supply-Chain Management (Gestão da Cadeia de Fornecimento) [42] são sistemas integrados destinados à gestão do negócio de fornecimento de produtos e serviços aos seus consumidores.

Um sistema *SCM* automatiza os processos envolvidos na produção e entrega de produtos e serviços, desde a procura da matéria-prima até à distribuição do produto final. Permite automatizar a cadeia de fornecimento, monitorizar e trocar informação relativa a pedidos e distribuições de/para parceiros e fornecedores. Desta forma é possível conduzir o negócio com maior velocidade e qualidade, prever as exigências do mercado e fornecer de forma mais exacta, permitindo planejar o negócio em torno das exigências dos clientes ao invés de planejar em função das capacidades de produção.

A implementação de um sistema *SCM* tem como objectivo produzir apenas o que é, quanto é, quando é e onde é necessário, reduzindo os níveis de stock e mantendo uma margem de segurança, garantindo a disponibilidade aquando da sua solicitação.

Os principais benefícios da sua implementação são a melhoria do desempenho na entrega de produtos a tempo e horas, os baixos níveis de *stock*, e uma maior sensibilidade para gestão de imprevistos, tais como falhas mecânicas, ausência de pessoal, atrasos nas entregas, recepção de produtos errados ou defeituosos, extravios, pedidos urgentes e erros humanos.

2.1.4 *PLM - Product Lifecycle Management*

Product Lifecycle Management (Gestão do Ciclo de Vida de Produtos) [33] são sistemas de gestão para todo o processo relacionado com o ciclo de vida dos produtos desde a sua concepção, passando pelo seu desenho e manufacturação, até à disponibilização no mercado.

Um sistema *PLM* engloba quatro áreas importantes: gestão do produto e do portfólio; desenho do produto; planeamento da manufacturação; e a gestão da informação do produto.

A implementação de um sistema *PLM* permite à organização desenvolver, descrever, gerir e comunicar informação acerca dos seus produtos. Auxilia a criação e

distribuição de produtos inovadores que preenchem ou criam as exigências do mercado e optimizam os processos de desenvolvimento de produtos e sistemas acelerando desta forma o seu lançamento no mercado, garantindo satisfação para a indústria, qualidade e cumprimento das normas reguladoras.

Estes sistemas permitem ainda tornar as organizações mais ágeis que as suas competidoras e aptas para reagirem e tomarem vantagem do mercado e das oportunidades competitivas através da rede de negócio.

2.1.5 *SRM - Supplier Relationship Management*

Supplier Relationship Management (Gestão do Relacionamento com o Fornecedor) [45] são sistemas de gestão dos processos de relacionamento entre organizações compradoras e fornecedoras de bens e serviços. Habitualmente estes sistemas estão integrados com sistemas *ERP*, *CRM*, *SCM* e *PLM* para poderem tirar partido de todos os dados gerados e centralizados pelos mesmos.

Os sistemas *SRM* são desenhados para apoiar os negócios em cada fase do ciclo de aquisição, disponibilizando ferramentas de selecção de fornecedores com base num conjunto de critérios, tais como preços e históricos de performance.

O principal objectivo destes sistemas é tornar mais eficientes os processos de interacção entre as empresas e os fornecedores, facilitando a negociação das aquisições e tornando-a mais eficiente.

2.2 Soluções Aplicacionais Integradas Existentes

Existem actualmente múltiplas soluções aplicacionais integradas no mercado que concretizam os sistemas de informação mencionados na anterior secção.

A implementação destes sistemas aplicacionais requer tanto conhecimento técnico como conhecimento do negócio a que se destina, sendo normalmente efectuada por empresas especializadas.

O processo de implementação inclui uma fase de parametrização e personalização dos módulos a instalar, de forma a responder às necessidades e ao contexto organizacional. A parametrização consiste num conjunto de passos bem definidos, constantes na documentação que acompanha habitualmente estes sistemas, onde são

fornecidos dados específicos da organização e definidas no sistema aplicacional as regras de funcionamento e a lógica de negócio.

São de seguida apresentadas algumas soluções comerciais de sistemas aplicacionais integrados.

2.2.1 Oracle E-Business Suite

A Oracle Corporation [25] é uma empresa mundial que desenvolve e comercializa o sistemas de gestão de base de dados Oracle Database [17] assim como o conjunto de aplicações de gestão de negócio empresarial o Oracle E-Business Suite [20].

O Oracle E-Business Suite é composto por cerca de 150 aplicações totalmente integradas que operam através da Internet e são suportadas por uma base de dados Oracle, destinada a servir de repositório de dados único e central. Este conjunto de aplicações abrange os sistemas *ERP*, *CRM* e *SCM*, que proporcionam processos de gestão da informação e permitem conectar e informatizar as organizações usando um sistema que partilha uma base tecnológica única.

O Oracle E-Business Suite, na sua *release* (versão) 11.5.10, é um pacote extenso composto por cerca de 258 mil objectos entre os quais 22 mil tabelas de base de dados, 31 mil ecrãs, 5 mil formulários, 6 mil relatórios com múltiplos módulos aplicacionais onde se destacam os seguintes: financeiro; compras; contratos; gestão de desempenho corporativo; gestão de dados de clientes; gestão de relacionamento com o cliente; gestão de recursos humanos; gestão de treino; logística; manutenção; manufacturação; marketing; gestão do ciclo de vida de produtos; vendas; serviços; acompanhamento de ordem de serviços junto dos fornecedores; planeamento de fornecedores; gestão de fornecedores; gestão de transportes.

Este conjunto de aplicações permite o acesso à informação de forma mais rápida e integrada automatizando os processos de negócios na organização. Integrando ferramentas de *Business Intelligence* é possível a transformação dos dados armazenados em informação de gestão, possibilitando a optimização do processo de tomada de decisão.

A Oracle disponibiliza igualmente um conjunto de ferramentas de desenvolvimento, o Oracle Developer Suite (iDS) [18]. Estas ferramentas permitem construir e estender

aplicações que podem ser disponibilizadas por múltiplos canais tais como portais, *Web Services* e dispositivos móveis.

O Oracle Developer Suite está dividido em módulos de ferramentas, dos quais se destacam o Oracle Forms [21], Oracle Reports [24], Oracle Discoverer [19] e o Oracle JDeveloper [23].

O Oracle Forms inclui o Oracle Forms Developer é uma linguagem de quarta geração que consiste numa ferramenta de desenvolvimento de aplicações com interfaces gráficos baseados em janelas e formulários para bases de dados Oracle, utilizando como linguagem de programação o PL/SQL.

O Oracle Reports inclui o Oracle Reports Developer e é igualmente uma linguagem de quarta geração que consiste numa ferramenta de desenvolvimento de relatórios sobre informação armazenada numa base de dados Oracle.

Para complementar a extracção de informação dos dados, existe o Oracle Discoverer que consiste num conjunto de ferramentas que, depois de parametrizadas, facilitam a criação de consulta *ad-hoc*, execução de relatórios, análise de dados e publicação *Web*, a utilizadores não técnicos.

O Oracle JDeveloper é um ambiente de desenvolvimento integrado (IDE) que tem um conjunto de características que permitem personalizar e desenvolver aplicações nas linguagens Java, XML, SQL e PL/SQL, HTML, JavaScript e PHP. Este módulo integra-se com o Oracle Application Development Framework (Oracle ADF) [16], uma *framework* Java criada pela Oracle Corporation, para o desenvolvimento e extensão de aplicações Oracle E-Business Suite.

As aplicações, formulários e relatórios desenvolvidos, podem ser disponibilizados através da *Web* utilizando para esse efeito o Oracle Internet Application Server (iAS) [22], que consiste numa colecção de serviços que possibilitam o acesso a estas aplicações e *Web Services* pela Internet oferecendo integração total com a plataforma J2EE.

As abordagens acima apresentadas, para criação e extensão de formulários, relatórios e aplicações, requerem um conhecimento profundo, por parte do programador, relativo às ferramentas e *APIs* (*Application Programming Interface*) disponibilizadas

para o seu desenvolvimento, exigindo que, para a apresentação rápida de resultados, seja necessária bastante experiência e familiarização contextual.

2.2.2 SAP Business Suite

A companhia alemã SAP AG [40] desenvolve e comercializa o pacote de soluções para gestão empresarial SAP Business Suite [36], que consiste num conjunto de aplicações de *software* que abrangem os sistemas *ERP*, *CRM*, *SCM*, *PLM* e *SRM*.

Destaca-se de entre este conjunto o SAP ERP, o sucessor do SAP R/3, que é considerado o principal produto da SAP. Este sistema aplicacional *ERP* oferece um conjunto de módulos com diversas aplicações de gestão de negócio. Os módulos são integrados e contêm a maior parte das funcionalidades necessárias às grandes organizações, incluindo manufaturação, finanças, vendas, distribuição de recursos humanos, entre outras.

2.2.3 Outras Soluções Aplicacionais Integradas

São apresentadas de seguida outras soluções aplicacionais integradas com forte incidência no mercado nacional.

PHC Enterprise

A companhia portuguesa PHC Software [32] comercializa o *ERP* PHC Enterprise [31] que consiste numa solução aplicacional integrada para a gestão de toda a informação relativa aos processos de negócio, de médias e grandes empresas.

Esta solução é baseada na tecnologia cliente/servidor, tendo como base de dados relacional o Microsoft SQL Server. Esta tecnologia assegura a elevada fiabilidade dos dados assim como a capacidade de crescimento para elevados fluxos de informação.

Primavera

A companhia portuguesa Primavera BSS (Business Software Solutions) [35] comercializa, entre outros produtos, o ERP Primavera [34] que consiste num conjunto de aplicações para gestão e optimização dos processos de negócio de uma organização.

Este produto integra um conjunto abrangente de módulos, totalmente integrados, que cobrem as principais áreas das empresas, permitindo assim uma fluidez de dados entre

as áreas financeira, logística, tesouraria, recursos humanos, etc. Neste sentido, é possível uma total integração de movimentos, baseada na utilização dinâmica de uma única base de dados, evitando a redundância de processos.

Microsoft Navision

O Navision [13] é um *ERP* comercializado pela Microsoft [15], que faz parte de um conjunto de aplicações, o Microsoft Dynamics [11], destinadas à gestão de todo o negócio das organizações relativo aos sectores financeiro, manufacturação, relacionamento com o cliente, cadeia de fornecimento, entre outros. Possui um gestor de base de dados nativo possibilitando também a integração com um Microsoft SQL Server.

2.2.4 Mercado dos Sistemas Aplicacionais Integrados

De toda a oferta existente no mercado relativa a produtos *ERP*, o SAP ERP é o produto que mais vendas realiza a nível nacional [38], ultrapassando os seus competidores mais directos, o da Microsoft Dynamics (Navision) e o da Oracle E-Business Suite. No que diz respeito a produtos *ERP* fabricados em Portugal, destacam-se o PHC e o Primavera, sendo o PHC o que obtém maior sucesso, com mais de 15 mil vendas, concorrendo com os líderes internacionais acima referidos.

A nível mundial a liderança é mantida pela companhia alemã SAP [39], sendo notória nos quatro segmentos de *software ERP*: sistemas de gestão financeira, gestão de capital humano, gestão de activos empresariais e operações de produção.

2.3 Plataforma de Desenvolvimento OutSystems

Nesta secção é descrita a plataforma de desenvolvimento OutSystems assim como uma solução desenvolvida através da mesma.

2.3.1 Plataforma OutSystems

A empresa portuguesa OutSystems [30] comercializa uma plataforma de desenvolvimento de *software* ágil que permite desenvolver, alterar e personalizar aplicações empresariais, disponibilizando interfaces de utilizador, acessíveis através da internet a partir de computadores pessoais e dispositivos móveis. Todo o ciclo de vida das aplicações é suportado e agilizado, garantindo que estas são criadas e

alteradas rapidamente para satisfazer as necessidades dos processos de negócio e permitir a entrega de aplicações em semanas, em vez de meses ou anos.

O nome da plataforma (OutSystems) está intrinsecamente associado ao seu objectivo principal que é o de oferecer uma plataforma de desenvolvimento ágil para a criação de aplicações em torno dos sistemas nucleares (*core systems*) implementados nas organizações.

Uma vez que o esforço e risco inerente ao estudo, planeamento e desenvolvimento de *core systems* é bastante elevado, e a complexidade de estender *core systems* é significativa, por requerer um conhecimento profundo da estrutura dos sistemas aplicacionais e das interfaces disponibilizadas para esse efeito, as organizações tendem a optar por desenvolver aplicações externas aos sistemas nucleares à medida das suas necessidades, e integrá-las com as já existentes.

Estas aplicações externas, devem reutilizar a lógica de negócio já implementada nos *core systems*, acrescentando funcionalidade e valor sem criar “ilhas” inacessíveis de informação.

A plataforma OutSystems é constituída por quatro componentes base que suportam toda a criação, alteração e manutenção de aplicações:

- **Service Studio:** É o ambiente visual de desenvolvimento de *eSpaces* (aplicações ou parte de aplicações) onde são totalmente desenhados, criados, modificados e testados, oferecendo abstracção a nível da codificação. É neste ambiente que são desenvolvidas as interfaces do utilizador e a lógica de negócio. Toda a consistência do *eSpace* é verificada antes de ser efectuado o *upload* para o servidor aplicacional (Hub Server), onde se dá a compilação e instalação tornando-o disponível para execução. Este IDE oferece um ambiente de desenvolvimento tecnologicamente independente permitindo que um *eSpace* seja transformado numa aplicação .NET ou Java, totalmente aberta e standard. Um *eSpace* construído por esta ferramenta é representado por um único ficheiro de extensão ‘.oml’;
- **Service Center:** É a consola de gestão e administração da plataforma OutSystems. É um ambiente intuitivo onde programadores, gestores aplicacionais e técnicos de

suporte, podem gerir o Hub Server. Esta oferece acesso centralizado a informação relativa a todos os recursos da plataforma tais como versões e gestão das aplicações, auditorias, *logging*, monitorização e criação de relatórios em tempo de execução;

- **Integration Studio:** É um ambiente que oferece um conjunto de funcionalidades que aceleram e automatizam o desenvolvimento de componentes integrados e extensões. Rápida e eficientemente podem ser criadas extensões (*Extension*) para serem integradas com sistemas principais (*core systems*) ou ainda em uso (*legacy systems*). Uma vez criada, uma *Extension* pode ser disponibilizada remotamente em qualquer Hub Server, incluindo todas as suas dependências, podendo ser consumida por qualquer *eSpace*. Uma *Extension* produzida por esta ferramenta é representada por um único ficheiro de extensão ‘.xif’;
- **Hub Server:** É o ambiente onde são orquestradas todas as acções de publicação, operação e gestão das aplicações. Tem como função alojar múltiplas aplicações para múltiplos serviços, distribuir múltiplas interfaces de utilizador para distintos perfis, integrar aplicações alojadas com sistemas internos ou externos, definir regras de acesso das aplicações ao nível da infra-estrutura e da lógica de negócio, assim como automatizar a disponibilização das aplicações. O Hub Server é operado centralmente por empresas ou prestadores de serviços para suportar o desenvolvimento colaborativo das aplicações empresariais.

A figura 1 [30] apresenta a arquitectura da plataforma OutSystems:



Figura 1 - Arquitectura OutSystems

A plataforma OutSystems está desenhada com grande foco na performance e escalabilidade. A sua arquitectura suporta um vasto conjunto de opções de instalação, variando desde departamentos com poucos utilizadores até grandes empresas suportando até um milhão de utilizadores. Esta arquitectura distribuída suporta balanceamento de carga (*load balancing*) pelos vários servidores (*clustering*), e remove os pontos únicos de falha no ambiente de execução.

Aquando da instalação da plataforma OutSystems, o utilizador pode optar pela base tecnológica (Java Enterprise Edition ou .NET) que melhor se adequa aos requisitos do seu negócio, podendo migrar para outra tecnologia assim que pretender, sem necessitar de reescrever a sua aplicação.

Uma aplicação poderá ser disponibilizada tanto numa plataforma constituída por um JavaServer Faces e linguagem Java usando um Java Application Server [8] em Linux

ou numa plataforma Microsoft ASP.NET e linguagem C# usando um Microsoft Internet Information Server [12] em Microsoft Windows Server.

A estrutura de armazenamento para suporte e disponibilização das aplicações pode ser uma base de dados Oracle ou Microsoft SQL Server [14].

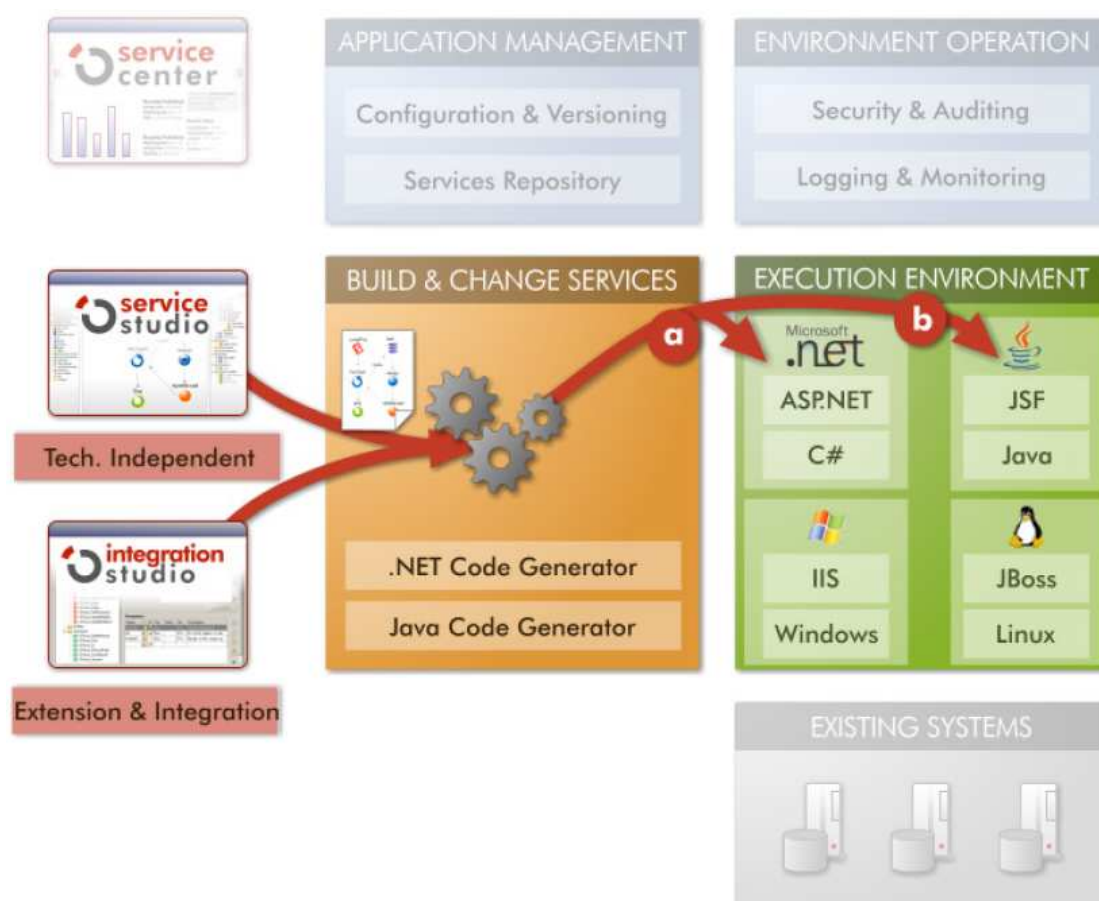


Figura 2 - Processo de Deploy (eSpace/Extension)

A figura 2 [30] ilustra o trajecto efectuado pelas aplicações e extensões até ao repositório de dados para finalmente serem disponibilizadas. Numa primeira fase, os *eSpaces* e extensões são criados através do Service Studio e Integration Studio, respectivamente, dando-se lugar a uma fase de geração de código (a) e disponibilização através da base tecnológica previamente definida (b).

A empresa OutSystems foi distinguida em 2006 pela Gartner ao fazer parte do Integrated Service Environment (ISE) Market Magic Quadrant [28]. Das 18 companhias incluídas, a OutSystems foi uma das três únicas europeias. Foram

avaliadas as características e funcionalidades dos produtos ISE, incluindo suporte a SOA e *Web Services* [47], desenvolvimento de portais e aplicações *Web*, influência na indústria e suporte à gestão de processos de negócio.

Mais recentemente, a OutSystems integrou a lista de finalistas nomeados, pela CMP Technology [4], para o 18º evento anual Dr. Dobb's Jolt Product Excellence Awards de 2008 [27], [9]. A OutSystems Platform 4.1 foi considerada como um dos seis melhores produtos no mercado mundial na categoria de Enterprise Tools. Os conceituados Jolt Awards são respeitados e conhecidos pelos seus processos de avaliação isentos, relativos à indústria do *software*. Destacam as empresas que causaram sensação com a inovação das suas ferramentas com capacidade de criarem mais rápida, fácil e eficientemente, *software*.

A OutSystems integrou também recentemente a lista de finalistas nomeados, pela Software & Information Industry Association (SIIA) [44], para o 23º evento anual Codie Awards de 2008 [5]. Os produtos em questão foram a plataforma OutSystems 4.1 para a categoria de Best On-Demand Platform e Best Software Development Solution, e o OutSystems Supplier Self Service [29] para a categoria de Best Supply Chain Solution.

No início de 2008 a OutSystems comunicou no seu site [26] o dado estatístico de mais de 10.000 *downloads* efectuados durante o ano de 2007, pelos seus utilizadores espalhados por mais de 130 países, relativamente ao seu pacote gratuito de funcionalidades limitadas, OutSystems Express Edition (versão gratuita da OutSystems Platform).

2.3.2 Agile Solutions for SAP

Sendo notória no mercado empresarial a presença de produtos SAP, a OutSystems apostou no desenvolvimento de um conjunto de soluções às quais deu o nome de Agile Solutions for SAP [1].

As Agile Solutions for SAP são um conjunto de aplicações que se destinam a organizações que pretendem estender e complementar os sistemas SAP existentes para melhorar continuamente o seu relacionamento com os empregados, clientes, fornecedores e parceiros de negócio.

Estas aplicações estão construídas sobre a *framework* Transparency Services for SAP que consiste num ambiente modular, aberto e flexível permitindo a disponibilização rápida de novas soluções de negócio que estendem e enriquecem os processos de negócio SAP. Todas as soluções podem ser totalmente integradas com as diferentes versões SAP, uma vez que esta *framework* usa toda a potencialidade da SAP Enterprise SOA [37]. A *framework* e soluções são modulares e completamente personalizáveis, adaptando-se facilmente aos requisitos em constante alteração e aos *standards* da companhia, para garantir que estão constantemente alinhados com o negócio.

A Transparency Services for SAP inclui um conjunto de módulos principais que disponibilizam funcionalidades nucleares podendo ser partilhadas pelas diferentes soluções de negócio. O objectivo principal é encapsular todas as interfaces com sistemas SAP e não-SAP, disponibilizando um conjunto de serviços de dados e serviços aplicacionais à camada superior, a Composite Applications.

A Composite Applications está focada nas soluções de negócio, podendo compreender tanto serviços SAP como não-SAP. Esta camada contém toda a lógica de negócio que suporta o conjunto de características disponíveis para os utilizadores finais e encapsula qualquer interface não *standard*. Estas aplicações compostas são baseadas nos *Enterprise Services (Web Services)* entregues pela SAP em conjunto com outras funcionalidades solicitadas.

Os serviços disponibilizados pela Transparency Services for SAP utilizam os SAP Enterprise Services e adicionam outros tais como *data caching*, sincronização, auditoria de negócio, gestão de utilizadores, mecanismos de fluxo de trabalho e notificação.

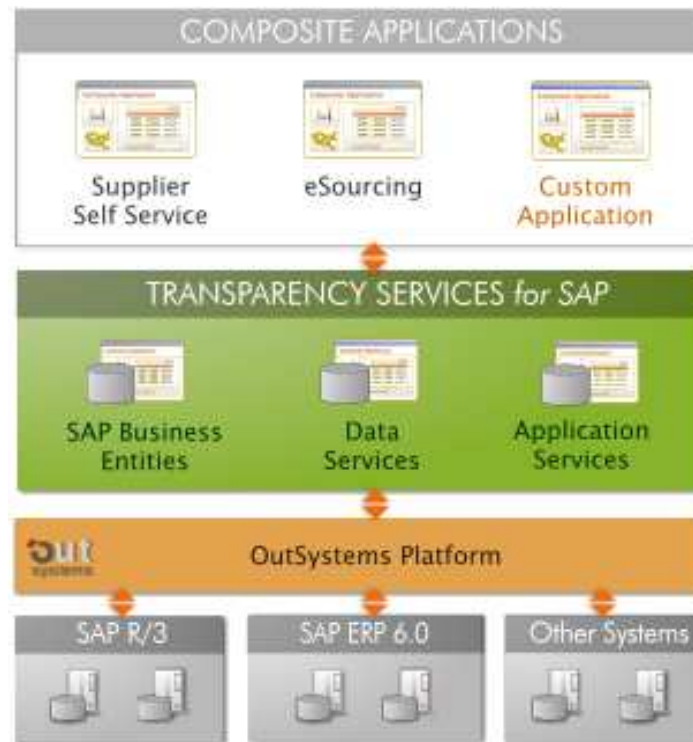


Figura 3 - Transparency Services for SAP

A figura 3 [1] apresenta a plataforma OutSystems, assente sobre os diversos sistemas aplicativos implementados, possibilitando que a camada Transparency Services for SAP disponibilize um conjunto de serviços de dados e serviços aplicativos à camada superior Composite Applications, a camada composta pelas aplicações periféricas desenvolvidas.

Capítulo 3

Metodologia e Planeamento

Neste capítulo são apresentadas as metodologias adoptadas para o desenvolvimento do projecto e é ainda realizada uma comparação e análise entre o plano previsto e o executado.

3.1 Metodologia de Desenvolvimento em Espiral

O desenvolvimento do Connector e da aplicação Generator, foi concretizado com a metodologia de desenvolvimento em espiral por se tratar de uma metodologia composta por um conjunto de fases sucessivas, que compõem cada ciclo do processo de desenvolvimento, até se atingir o objectivo traçado, e por ser apropriada para projectos onde se desconhecem à partida todos os detalhes das funcionalidades a criar, oferecendo prototipagem e flexibilidade de integração dos componentes desenvolvidos, após qualquer ciclo. Cada ciclo é composto por uma fase de Análise, Desenho, Implementação e Teste [2].

Com esta metodologia, na fase de análise é especificado um subconjunto de requisitos da aplicação por forma a serem traçados os objectivos para cada ciclo de desenvolvimento. Na fase seguinte é efectuado o desenho da aplicação, seguindo-se a fase de implementação onde é codificado e concretizado o desenho até então projectado. O ciclo termina com uma fase de avaliação do protótipo e teste de todos os componentes da aplicação desenvolvidos.

A figura 4 [10] ilustra o modelo de desenvolvimento em espiral.

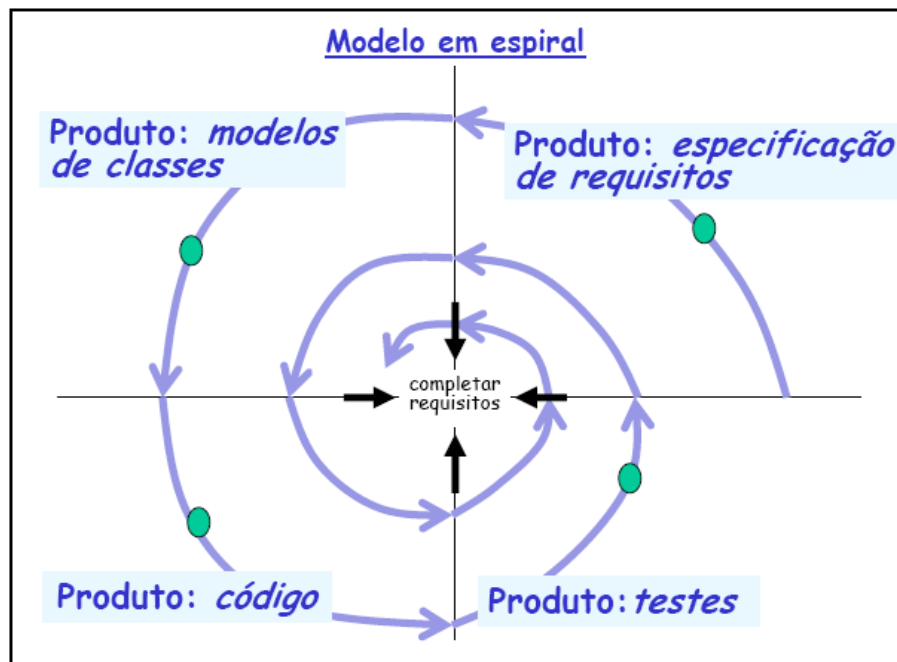


Figura 4 - Metodologia de desenvolvimento em Espiral

Durante cada ciclo de desenvolvimento, é possível obter versões da aplicação cada vez mais evoluídas e, recorrendo à prototipagem, é possível efectuar uma avaliação mais capaz reduzindo os riscos inerentes ao seu desenvolvimento.

Este modelo segue uma abordagem de refinamento, à semelhança de outros, e uma vez que é iterativo, reflecte realisticamente o processo de desenvolvimento.

Esta foi a metodologia de desenvolvimento adoptada para a concretização do Connector e da aplicação Generator tendo sido realizadas 3 iterações, das várias fases, bem como um refinamento apurado do funcionamento dos componentes desenvolvidos. O detalhe de execução das iterações é apresentado na secção Planeamento.

3.2 Metodologia de Desenvolvimento Scrum

Para a concretização da aplicação Receivables Workbench, foram adoptados os conceitos principais da metodologia Scrum [43] [41], uma metodologia de

desenvolvimento de projectos em equipa que tem sido aplicada com sucesso ao desenvolvimento ágil de *software*.

Esta é uma corrente que segue uma linha distinta das metodologias tradicionais, que por vezes têm falhado na execução de projectos de sistemas de informação. A engenharia de *software* é uma actividade de gestão de conhecimento que tem de lidar com múltiplas incertezas, sejam elas tecnológicas, de negócio, sociais ou outras. Todas estas incertezas aumentam os riscos dos projectos, levando a situações em que, para se minimizar estes riscos, são adoptadas metodologias que atrasam significativamente o *break-even* (ponto de equilíbrio) do projecto (seja por via da elaboração inicial de documentos de especificação muito detalhados, seja por via de atrasos no processo de desenvolvimento provocados por erros ou omissões nos documentos de especificação, seja por erros de estimação detectados em fases tardias do projecto).

A generalidade das metodologias em uso resiste à mudança, procurando adiar todas as alterações ao projecto para depois da execução do mesmo. Esta estratégia é diametralmente oposta às necessidades do negócio: ao longo do desenvolvimento do projecto o negócio quer continuar a alterar métodos e procedimentos para aumentar a competitividade da organização.

Scrum é uma metodologia ligeira e ágil para o desenvolvimento de *software* que conjuga as práticas de desenvolvimento de *software* mais utilizadas. É usado para gerir e controlar o desenvolvimento de *software* e de outros produtos usando uma prática iterativa e incremental, proporcionando ao dono do produto informação diária e quinzenal acerca do desenvolvimento do mesmo. É ideal para projectos com ritmos de mudança ou de incerteza elevados.

O Scrum foca-se em entregar rapidamente ao negócio as funcionalidades com maior valor, entregando pequenas parcelas de funcionalidade em ciclos reduzidos e aumentando a previsibilidade através da angariação de dados de estimação específicos do projecto e ajustando frequentemente o planeamento ao invés de apostar num plano completo que, ao falhar, poderá levar a resultados muito diferentes do esperado.

Introduzem-se de seguida noções e conceitos desta metodologia utilizados ao longo do documento bem como as principais vantagens da sua utilização.

3.2.1 Papéis

O Scrum tem como intervenientes e papéis fundamentais o *Product Owner* (dono do produto), o *Scrum Master* (gestor do Scrum) e a *Team* (equipa).

Product Owner

O *product owner* (dono do produto), proposto pela sua organização, representa e gere as agendas das partes interessadas no projecto transformando estas agendas no *product backlog*. O *product backlog* é uma lista de requisitos prioritizada pelo valor que cada um destes requisitos representa.

O *product owner* tem a responsabilidade de estabelecer, alimentar e comunicar a visão do produto, sendo o evangelista do produto junto das partes interessadas e, em particular dos utilizadores.

O *product owner* tem a responsabilidade de monitorizar o desenvolvimento do projecto face aos objectivos e visão de investimento, apurando em cada momento o *ROI - Return On Investment* (retorno de investimento) do projecto.

O *product owner* decide quando criar uma *release* oficial do produto.

A empresa de desenvolvimento de *software* colaborará com o *product owner* na redacção dos requisitos, mas será responsabilidade exclusiva do *product owner* a sua priorização, bem como a comunicação acerca do produto com as restantes partes interessadas.

O *product owner* deverá estar disponível para uma reunião periódica e para esclarecer prontamente a equipa de projecto acerca de quaisquer dúvidas que surjam no desenvolvimento do mesmo, e deverá ter pronto acesso a todos os recursos (pessoas, procedimentos, legislação ou outros) necessários ao esclarecimento destas dúvidas.

Scrum Master

O papel do *scrum master* (gestor do Scrum) é manter uma equipa de desenvolvimentos saudável, gerindo a reunião de progresso diária, protegendo a equipa da desfocagem nos objectivos, eliminando todas as questões que possam bloquear o progresso da equipa, coordenando os pedidos de alocação de tempo de membros da equipa para outras tarefas, evangelizando a equipa, o *product owner* e as partes interessadas para os valores e princípios subjacentes ao Scrum.

O *scrum master* é responsável pela produtividade da equipa e define e reporta ao *product owner* esta produtividade. Caso o *product owner* consiga valorizar cada um dos requisitos presentes no *product backlog* o *scrum master* conseguirá também valorizar objectivamente a produtividade da equipa.

Cabe à empresa de desenvolvimento do projecto nomear um *Certified Scrum Master* para desempenhar o *role* de *scrum master*.

Team

A *team* (equipa) executa o *product backlog* de acordo com a visão definida para o produto. A *team* define quais os compromissos que está disposta a assumir perante o *product owner* e gere o seu próprio trabalho por forma a atingir os compromissos assumidos.

A *team* é multi-funcional, fazendo o que for necessário para completar as tarefas, desenvolvendo sempre em primeiro lugar as funcionalidades mais prioritárias do *product backlog*.

3.2.2 *Sprinting*

Scrum é uma metodologia iterativa com base no conceito de *sprints*. Um *sprint* é um período de tempo reduzido (tipicamente 2 semanas) durante o qual a equipa assume o compromisso de entregar um conjunto de funcionalidades demonstráveis, implementar essas funcionalidades e demonstrá-las no último dia do *sprint*.

Planning Meeting 1

O *sprint* inicia-se com uma reunião de planeamento em que a equipa decide (com base no *product backlog*, nas prioridades atribuídas pelo *product owner*, nas estimativas definidas pela equipa e nos recursos disponíveis na equipa) o que será construído no *sprint*.

O *product owner* participará nesta decisão e procurará influenciar a equipa direccionando para a construção de funcionalidades com maior valor.

Planning Meeting 2

A equipa realizará uma segunda reunião em que decidirá como construir as funcionalidades seleccionadas na *Planning Meeting 1*.

É criado um *sprint backlog* em que as funcionalidades a construir são detalhadas e partidas em múltiplas tarefas. O *sprint backlog* é inicializado e configurado para suportar o *sprint*.

Nesta reunião é expectável que a equipa se auto-organize, onde os membros da equipa se voluntariam para realizar as tarefas inscritas no *sprint backlog*.

Daily Meeting

Diariamente a equipa realiza uma breve reunião de estado em que cada membro reporta à equipa o que fez no último dia, o que planeia fazer no próximo dia e quais os impedimentos que bloqueiam o seu progresso.

Esta reunião promove a orquestração do trabalho entre os vários membros da equipa, criando um ponto de controlo diário que serve para aumentar a visibilidade sobre problemas inesperados na equipa e/ou no trabalho.

O *scrum master* modera esta reunião anotando todos os impedimentos que deverá remover, avaliando também o progresso da equipa face aos objectivos estabelecidos no *sprint backlog*. Faz parte da moderação levar a discussão questões técnicas ou quaisquer problemas particulares entre membros da equipa.

Demo Session

No último dia do *sprint* a *team* encenará uma *demo session* em que demonstrará ao *product owner* as funcionalidades criadas durante o *sprint*. Nesta *demo session* o *scrum master* apresentará ao *product owner* detalhes acerca do progresso realizado pela *team* durante o *sprint*, incluindo gráficos que permitem analisar as funcionalidades implementadas face à capacidade de implementação instalada.

O *product owner* reagirá à *demo* alimentando a equipa com *feedback* importante para melhorar as funcionalidades demonstradas.

Team Retrospective

A última actividade do *sprint* é uma análise retrospectiva, analisando-se o que correu bem e o que correu mal durante o *sprint*, inspeccionando-se os aspectos relevantes e adaptando-se acções correctivas a implementar no *sprint* seguinte.

O *product owner* participará activamente nesta reunião.

3.2.3 Artefactos

Os artefactos utilizados durante o processo de desenvolvimento do projecto são o *product backlog* e os *sprint backlogs*.

Product Backlog

O *product backlog* é uma lista prioritizada de funcionalidades e outros trabalhos necessários à implementação do projecto.

O *product owner* define os itens que constam nesta lista e prioriza-os. Tanto o *product owner*, como o *scrum master* como qualquer membro da equipa pode adicionar elementos ao *product backlog* em qualquer altura, no entanto é responsabilidade exclusiva do *product owner* a priorização dos itens constantes do *product backlog* e é responsabilidade exclusiva da equipa a estimação grossa destes itens.

O *product backlog* poderá conter funcionalidades, defeitos, oportunidades de melhoria ou pré-condições (trabalhos preparatórios necessários à realização de funcionalidades que criem valor).

O critério a adoptar pelo *product owner* para a priorização dos itens constantes no *product backlog* deverá ser o do valor acrescentado: são mais prioritários e serão desenvolvidos primeiro, os itens com maior valor acrescentado. Este critério permitirá colocar em produção uma versão parcial do projecto, em que, apesar de não estarem implementadas todas as funcionalidades, estão implementadas as funcionalidades que maior valor aportam ao projecto. Isto permitirá começar a realizar valor com base no projecto mais cedo, antecipando o *break-even* e aumentando o *ROI*.

Sprint Backlog

O *sprint backlog* é uma lista prioritizada de tarefas necessárias à implementação das funcionalidades seleccionadas para implementação durante o *sprint*.

A equipa constrói o *sprint backlog* através da análise e detalhe das funcionalidades seleccionadas, identificando todas as tarefas necessárias para implementar a funcionalidade.

O *sprint backlog* é actualizado diariamente pela equipa e permite acompanhar o progresso do *sprint*, permitindo aferir, logo a partir do terceiro dia do *sprint*, as probabilidades de completar os objectivos do *sprint*.

A qualquer momento qualquer membro da equipa pode adicionar itens ao *sprint backlog*, sendo apenas autorizada a inclusão de itens que possam ser considerados tarefas necessárias para a realização das funcionalidades do *sprint*.

A figura 5 [41] ilustra a mecânica da metodologia Scrum aqui descrita.

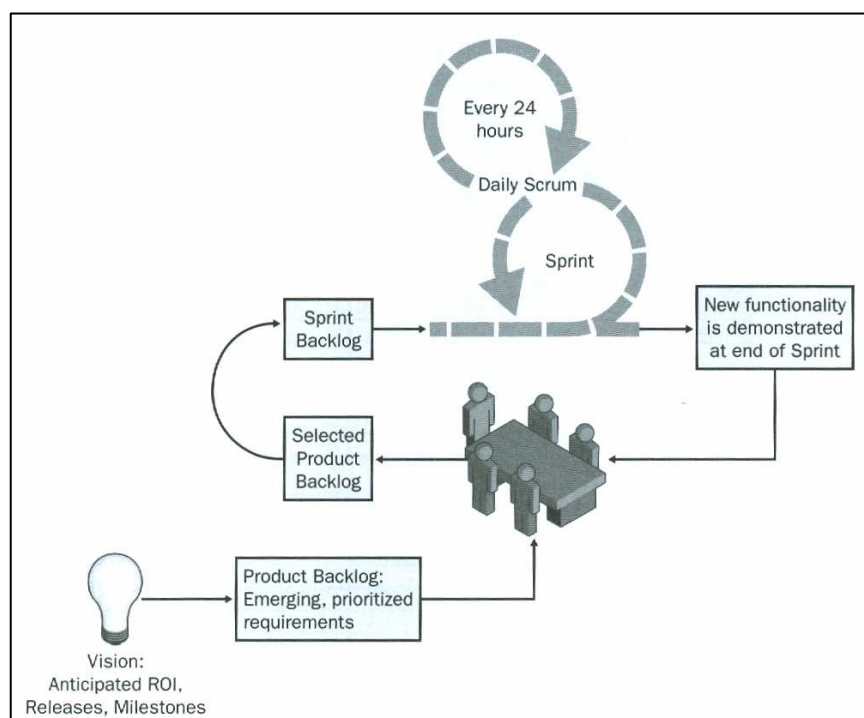


Figura 5 - Metodologia de desenvolvimento Scrum

3.2.4 Discussão

A aplicação da metodologia Scrum no desenvolvimento de *software* tem múltiplas vantagens e desvantagens, realçando-se as seguintes:

- O projecto de *software* entrega muito rapidamente ao negócio as funcionalidades com maior valor, permitindo antecipar o *break-even* do projecto e aumentar o *ROI*;

- O negócio pode, a qualquer momento, modificar, cancelar ou incluir novas funcionalidades no projecto sendo o custo destas alterações virtualmente nulo sempre que as mesmas versem áreas do projecto ainda não implementadas;
- O negócio tem total visibilidade do projecto, sendo evidentes os custos relacionados com a inclusão de modificações que versem áreas do projecto já implementadas; caso decida alterar estas áreas os custos da alteração são relativamente reduzidos quando comparados com as metodologias tradicionais;
- O negócio pode cancelar o desenvolvimento do projecto sempre que considere que o valor acrescentado dos itens pendentes no *product backlog* não justifica a sua implementação (cenário típico numa fase adiantada de projecto em que todas as funcionalidades importantes foram implementadas e em que restam apenas no *product backlog* itens de baixo valor), podendo o negócio ajustar-se rapidamente a cenários de contracção em que a contenção de custos obrigue a suspender o projecto, sem perder o investimento entretanto realizado na sua implementação;
- É fundamental que os intervenientes conheçam e tenham prática no emprego desta metodologia, respeitando as regras e utilizando os artefactos necessários, para não inviabilizar o desenvolvimento do projecto;
- Não é indicado para o desenvolvimento de projectos de pequena dimensão, sendo necessários no mínimo 3 elementos (*product owner*, *scrum master* e *team*);
- Por ser uma metodologia recente, diferente, e por atribuir responsabilidades, verifica-se resistência à mudança por parte dos intervenientes;
- Requer constante monitorização quer qualitativa quer quantitativa.

O Scrum não acrescenta novidades significativas relativamente às metodologias tradicionais e trata apenas de realçar num conjunto de práticas os aspectos cruciais para o sucesso da implementação de um projecto de *software*.

Esta foi a metodologia de desenvolvimento adoptada para a concretização da aplicação Receivables Workbench tendo sido, para este efeito, nomeado um *product owner* e um *scrum master*. Para o desenvolvimento foi elaborado um *product backlog* concretizado em 4 *sprints*.

3.3 Planeamento

Para o desenvolvimento do projecto, o planeamento inicial incluía a realização de tarefas relativas ao desenvolvimento do Conector OutSystems para o Oracle E-Business Suite, assim como um conjunto de módulos aplicativos para a gestão de fornecedores e facturas, constituintes do negócio de diversos sectores empresariais.

O mapa de Gantt da figura 6 ilustra o plano inicialmente previsto:

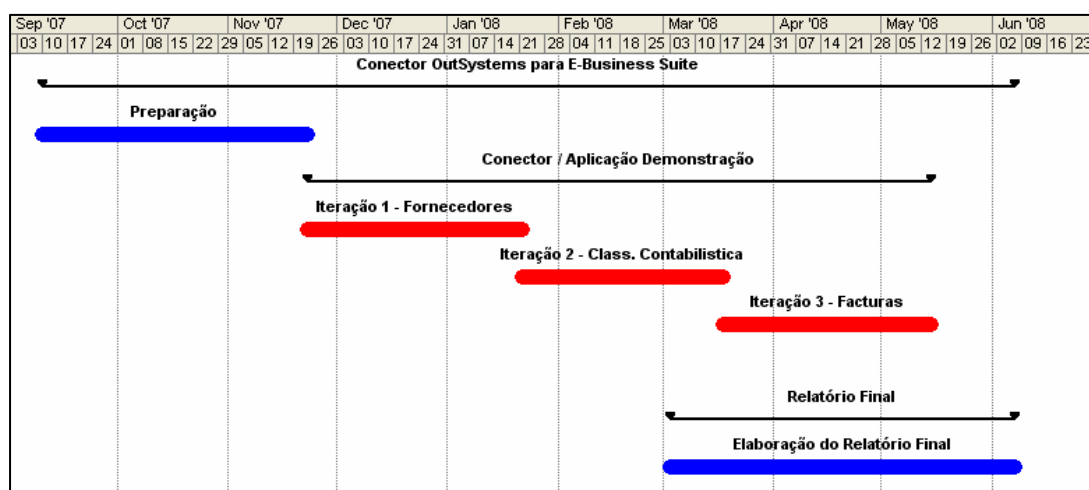


Figura 6 - Planeamento inicial

De acordo com o planeamento inicial, a fase 'Conector / Aplicação Demonstração' corresponde ao período de desenvolvimento do projecto propriamente dito, tendo este sofrido algumas alterações. Neste período e após uma fase de redefinição de objectivos, foram desenvolvidas as seguintes tarefas: Integração e Configuração - OEBS; Connector e Aplicação Generator; Aplicação Receivables Workbench.

A figura 7 apresenta a distribuição destas tarefas ao longo do período de execução do projecto.

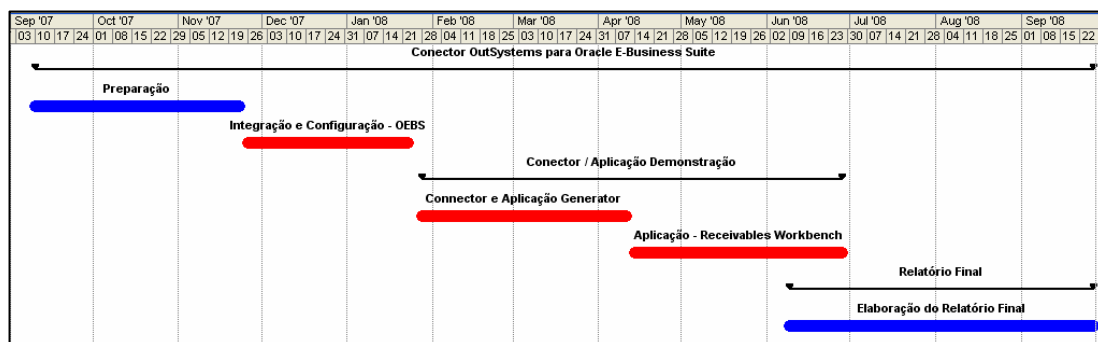


Figura 7 - Planeamento final

Estas alterações ao planeamento resultaram da redefinição conjugada de objectivos em função de uma oportunidade criada pelo início de um projecto na PRÁXIA, que potenciou o contexto ideal para o desenvolvimento deste projecto em ambiente real.

Em função desta redefinição e para a aquisição do conhecimento necessário ao estudo e desenvolvimento do conector OutSystems, foi seguido um trajecto profissional focado na tecnologia Oracle, nomeadamente Oracle Database e E-Business Suite. Este trajecto incluiu a participação na equipa de implementação e adaptação do Oracle E-Business Suite às necessidades de um cliente para o qual a PRÁXIA presta os seus serviços.

Após aquisição do conhecimento sobre a tecnologia Oracle, deu-se início ao estudo e concepção da solução Connector para integração com a plataforma OutSystems. Foram ainda nesta fase efectuados testes com vista à obtenção de resultados que indicassem o correcto funcionamento do conector gerado.

Foi iniciado de seguida, o projecto de construção da aplicação Receivables Workbench. Esta aplicação visa automatizar processos relacionados com a gestão financeira, nomeadamente, a gestão de receitas e respectivas transacções, armazenando e utilizando toda a informação disponível no repositório central e único do Oracle E-Business Suite.

A tabela 1 detalha as tarefas realizadas e a sua duração durante o projecto.

Tarefa	Dias	Data Início	Data Fim
Preparação	54 days	10-Set-07	22-Nov-07
Integração na Empresa (PRÁXIA)	5 days	10-Set-07	14-Set-07
Aprendizagem da Tecnologia OutSystems	28 days	17-Set-07	24-Out-07
Aprendizagem de Ferramentas / outras Tecnologias	11 days	25-Out-07	08-Nov-07
Análise e Elaboração do Relatório Preliminar	10 days	09-Nov-07	22-Nov-07
Integração e Configuração - OEBS	46 days	27-Nov-07	29-Jan-08
Instalação plataforma de suporte ao OEBS	21 days	27-Nov-07	25-Dez-07
Instalação / Configuração das aplicações	25 days	26-Dez-07	29-Jan-08
Connector e Aplicação Generator	56 days	30-Jan-08	16-Abr-08
Ciclo 1	23 days	30-Jan-08	29-Fev-08
Análise	5 days	30-Jan-08	05-Fev-08
Desenho	5 days	06-Fev-08	12-Fev-08
Implementação	10 days	13-Fev-08	26-Fev-08
Teste	3 days	27-Fev-08	29-Fev-08
Ciclo 2	20 days	03-Mar-08	28-Mar-08
Análise	3 days	03-Mar-08	05-Mar-08

Tarefa	Dias	Data Início	Data Fim
Desenho	4 days	06-Mar-08	11-Mar-08
Implementação	10 days	12-Mar-08	25-Mar-08
Teste	3 days	26-Mar-08	28-Mar-08
Ciclo 3	13 days	31-Mar-08	16-Abr-08
Análise	2 days	31-Mar-08	01-Abr-08
Desenho	3 days	02-Abr-08	04-Abr-08
Implementação	5 days	07-Abr-08	11-Abr-08
Teste	3 days	14-Abr-08	16-Abr-08
Aplicação - Receivables Workbench	51 days	17-Abr-08	26-Jun-08
Sprint 1	14 days	17-Abr-08	06-Mai-08
Sprint 2	9 days	07-Mai-08	19-Mai-08
Sprint 3	14 days	20-Mai-08	06-Jun-08
Sprint 4	14 days	09-Jun-08	26-Jun-08
Relatório Final	80 days	09-Jun-08	26-Set-08
Elaboração do Relatório	80 days	09-Jun-08	26-Set-08

Tabela 1 - Planeamento final detalhado

Na iteração 1 do desenvolvimento do Connector e aplicação Generator foram realizados os módulos de recepção e interpretação dos procedimentos/funções a serem gerados, e foi identificado e implementada a solução para o problema de compatibilidade entre os tipos de dados manipulados pelas tecnologias OEBS e OutSystems/.NET.

Na iteração 2 foram realizados os módulos de geração e composição dos ficheiros gerados, e identificado e solucionado o problema da não representatividade do valor 'null' por parte da tecnologia OutSystems.

Por último, na iteração 3, foi automatizado o processo de geração e disponibilização dos ficheiros gerados e compilados pela aplicação Generator, obtendo-se o conjunto de métodos de conexão que compõem o Connector.

No Capítulo 5 são desenvolvidos e apresentados detalhes da execução destas actividades.

Para o desenvolvimento da aplicação Receivables Workbench foi seguida a metodologia Scrum, sendo esta caracterizada pela utilização de artefactos da metodologia como o *product backlog* e *sprint backlog*. Durante este período, foram realizados 4 *sprints* tendo cada um a duração de duas semanas, aproximadamente.

Para ilustrar a mecânica da metodologia adoptada, a figura 8 ilustra um dos *sprint backlogs* definidos durante o desenvolvimento da aplicação, contendo um subconjunto das funcionalidades do *product backlog*.

[illegible]

Figura 8 - Sprint Backlog

Este *sprint* teve a duração de duas semanas, após uma reunião para definição dos objectivos a alcançar. O registo das funcionalidades foi efectuado no *sprint backlog*, sendo este actualizado diariamente, à medida que as tarefas foram executadas.

São ainda evidentes nesta lista a prioridade de cada funcionalidade (definida pelo *product owner*), a duração dispendida e o estado final de cada tarefa (completa / cancelada / pendente / por iniciar).

A figura 9, por sua vez, apresenta um resumo e uma análise da execução do *sprint backlog* representando a capacidade associada aos programadores intervenientes, o esforço dispendido e o progresso total das tarefas, através de gráficos ilustrativos.

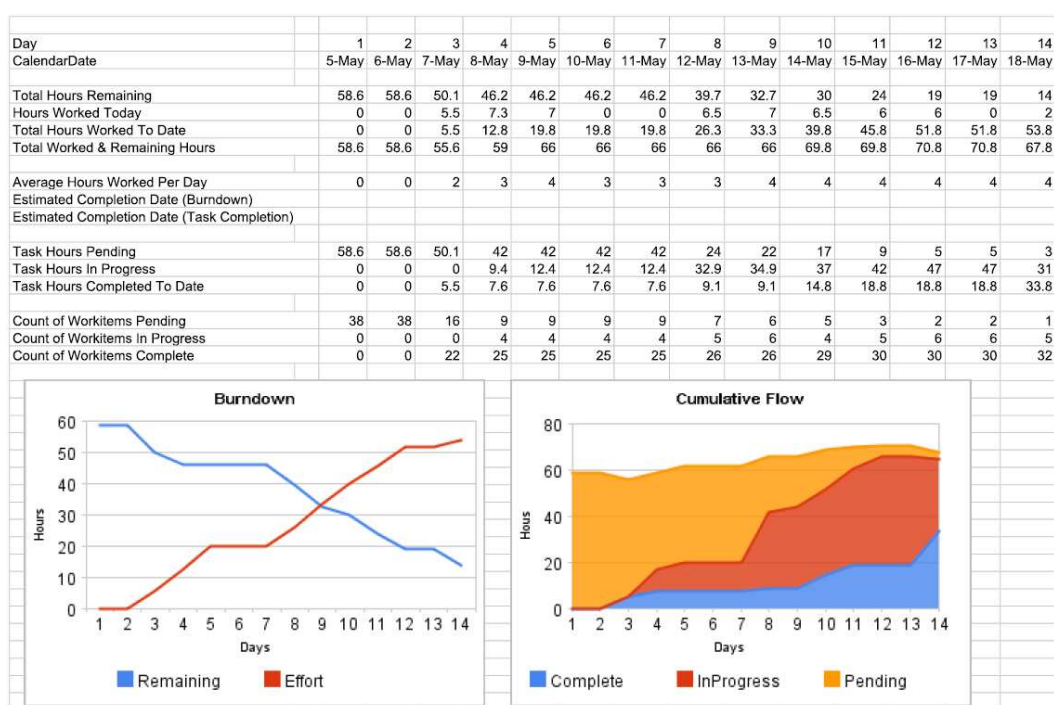


Figura 9 - Análise do *Sprint Backlog*

Nesta imagem o gráfico *Burndown* representa o número de horas relativas às tarefas a realizar (*Remaining*) e o esforço relativo ao desenvolvimento das mesmas (*Effort*), durante o *sprint*. No gráfico *Cumulative Flow* podemos verificar gradualmente a evolução das tarefas, nomeadamente o número de horas completadas (*Complete*), em progresso (*InProgress*) e pendentes (*Pending*).

Com base nesta informação, é possível obter a percepção da evolução dos trabalhos, reagir às ocorrências do projecto por forma a minimizar os desvios e efectuar uma análise e discussão mais capaz nas reuniões diárias (*daily meeting*) e reuniões de avaliação e conclusão dos *sprints* (*team retrospective*).

Capítulo 4

Arquitectura do Sistema

A arquitectura do sistema espelha as camadas de interface propostas e fundamentais para permitir a comunicação entre as tecnologias OutSystems e Oracle E-Business Suite.

Nesta secção são realçadas as principais camadas desta arquitectura, destacando-se a OEBS Transparency Services e Composite Applications que representam, respectivamente, a camada de serviços necessários à comunicação e manutenção das aplicações desenvolvidas através da tecnologia OutSystems que interagem com o Oracle E-Business Suite, e as aplicações desenvolvidas/a desenvolver com esta infraestrutura.

4.1 Arquitectura Global

A arquitectura geral do sistema é ilustrada pelo diagrama da Figura 10. Neste diagrama podemos observar cinco níveis distintos, representando as múltiplas camadas de uma instância da plataforma OutSystems assente sobre um sistema Oracle E-Business Suite e a comunicação com o componente de conexão gerado. As camadas são: Sistemas Base, Plataforma OutSystems, Generator, OEBS Transparency Services e Composite Applications.

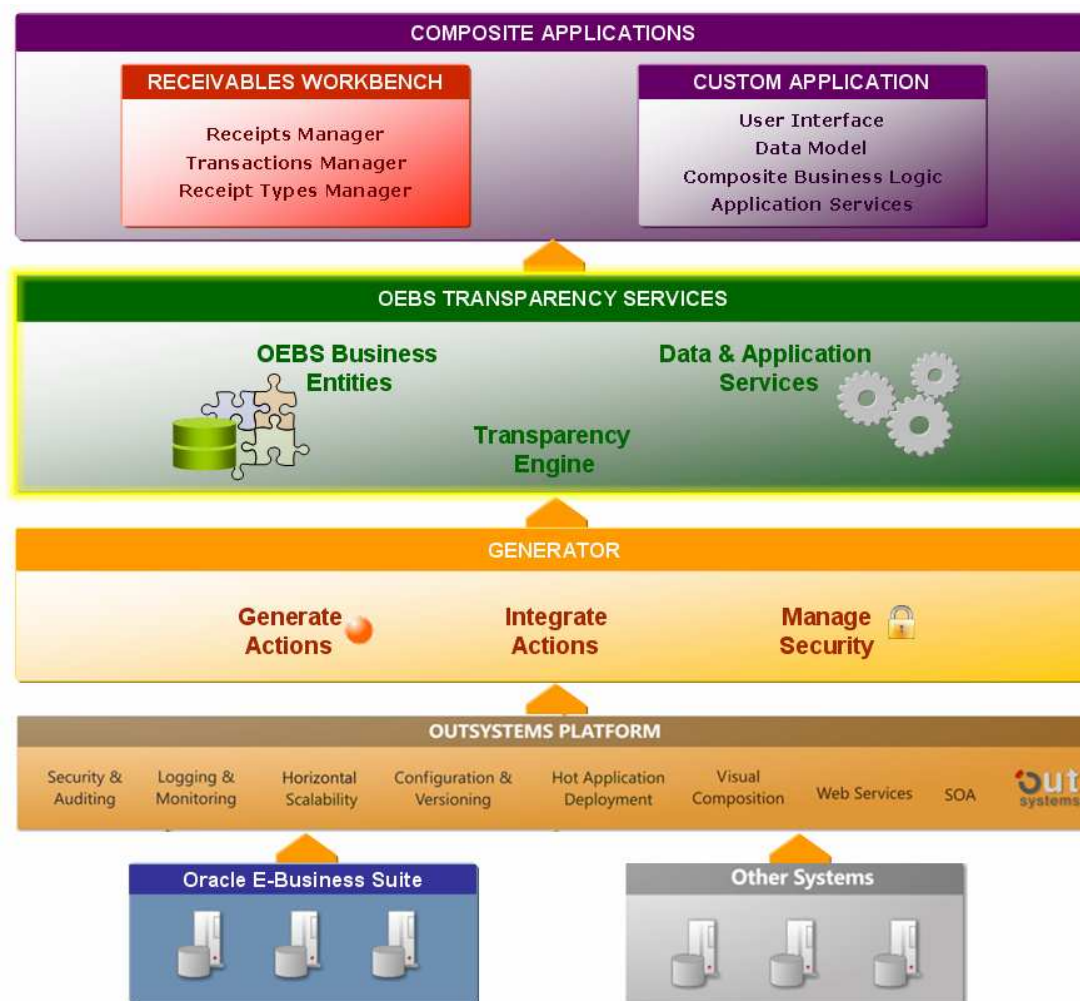


Figura 10 - Arquitectura Global do Sistema

De seguida são descritas com maior detalhe as diferentes camadas da arquitectura envolvidas na execução do projecto.

4.1.1 Sistemas Base

A primeira camada (nível inferior) representa os sistemas nucleares com que poderá interagir uma instância da plataforma OutSystems.

Neste caso concreto do projecto, a plataforma OutSystems interage sobre um sistema aplicacional Oracle E-Business Suite (OEBS). Desta forma é possível ter acesso a toda a informação e lógica de negócio nele constante permitindo assim uma correcta execução e exploração da estrutura de suporte do OEBS.

A figura 11 representa a arquitectura geral de um sistema OEBS:

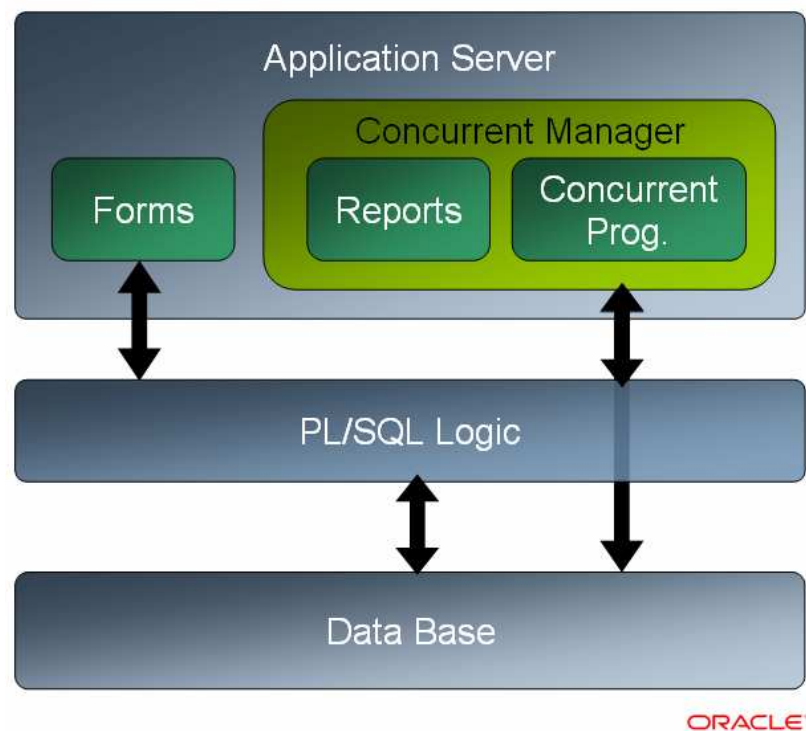


Figura 11 - Arquitectura Geral OEBS

A arquitectura geral do OEBS inclui uma base de dados, um conjunto de funções de negócio implementadas em PL/SQL e uma camada aplicacional constituída por três grandes componentes:

- Formulários (*Forms*) que invocam a lógica de negócio no PL/SQL e implementam também lógica aplicacional de controlo;
- Relatórios (*Reports*) e programas concorrentes (*Concurrent Programs*) que, directamente sobre a base de dados ou através da lógica de negócio em PL/SQL, recolhem, apresentam e manipulam informação;
- Motor de processamento concorrente que gere as prioridades e execução de relatórios e programas.

O OEBS possibilita a interacção com aplicações externas através de dois grandes tipos de interfaces:

- O uso de tabelas de interface que, conjugadas com a execução de programas concorrentes de importação e validação, possibilitam a introdução de dados no sistema. Este é o método clássico;
- *APIs* seleccionadas em PL/SQL, que apesar de limitadas para a generalidade das funções de negócio, fornecem o acesso necessário para a criação e manipulação das principais entidades do negócio. Este é um método mais evoluído mas ainda muito limitado.

No desenvolvimento do projecto foram estudadas e utilizadas estas duas formas de interacção com o OEBS como detalhado no Capítulo 5.

4.1.2 Plataforma OutSystems

A segunda camada representa a plataforma OutSystems. Esta oferece um conjunto de mecanismos tais como auditoria, monitorização, escalabilidade entre outros, permitindo o desenvolvimento e disponibilização de aplicações.

A figura 12 representa a arquitectura OutSystems:

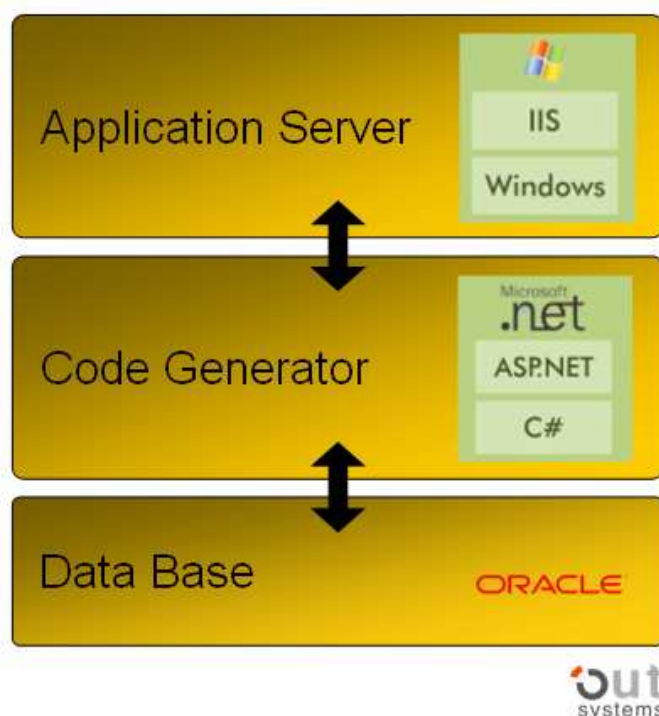


Figura 12 - Arquitectura da Instância da Plataforma OutSystems do Projecto

A plataforma OutSystems proporciona uma abstracção do processo de desenvolvimento de *software* disponibilizando, através de uma ferramenta proprietária, a codificação de acções e fluxos através de uma representação gráfica. Há um conjunto de acções e fluxos já embebidos na plataforma que possibilitam a criação das funcionalidades básicas de uma aplicação. Para atingir este objectivo recorre a um conjunto de componentes que envolvem:

- Base de dados: onde armazena e gere o conjunto de objectos relacionais das aplicações bem como outras entidades responsáveis pelo controlo da plataforma;
- Código compilado C# .NET: que implementa as funções, acções, primitivas e fluxos representados graficamente pela plataforma e que, conjugados pela codificação gráfica na ferramenta de desenvolvimento, concretizam as aplicações;
- Servidor aplicacional IIS: que interpreta e executa o código que foi preparado para as aplicações dando o contexto aplicacional necessário em .NET.

O conjunto de acções, primitivas e outras funções disponibilizadas é bastante amplo. No entanto, esta plataforma não está preparada para disponibilizar, de uma forma automática, a lógica de negócio dos sistemas aplicacionais OEBS, limitando o desenvolvimento ágil de aplicações externas ao OEBS.

A forma disponibilizada para estender a plataforma a novas funções de negócio que sejam incluídas na ferramenta de desenvolvimento, envolve a codificação em .NET das funções pretendidas e posterior integração na ferramenta através de uma biblioteca *DLL (Dynamic-Link Library)*. Esta foi a forma utilizada para estender a plataforma para incluir funções próprias dos sistemas base e para garantir a ligação com o OEBS através de um conjunto de componentes.

4.1.3 Generator

A camada seguinte diz respeito à aplicação Generator. Desenvolvida através da tecnologia OutSystems e linguagem C# .NET, esta é responsável por gerar os componentes de conexão da plataforma com o OEBS, tornando-se parte constituinte da camada acima, a OEBS Transparency Services.

A figura 13 representa a camada Generator do sistema implementado:

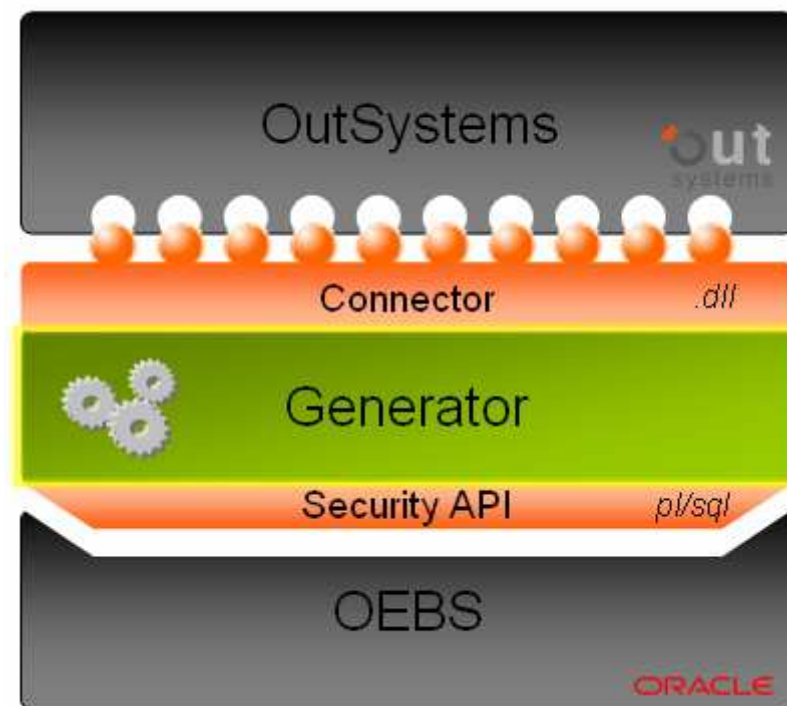


Figura 13 - Representação gráfica do componente Generator - conexão entre a plataforma OutSystems e o OEBS

A camada/aplicação Generator explora a lógica de negócio do OEBS e gera os pontos de conexão desejados, sendo possível a extensão ou criação de novas aplicações em redor das já oferecidas pelos sistemas nucleares. Para o concretizar é gerada uma biblioteca em .NET que inclui a implementação das funções de negócio seleccionadas e que, depois de integrada na ferramenta de desenvolvimento, permitirá a selecção das acções disponibilizadas para inclusão nas aplicações a desenvolver.

Esta camada é também responsável por construir um componente de segurança que controla o acesso aos recursos da base de dados. Este componente cria uma camada de interface junto da base de dados impedindo o acesso directo às APIs disponibilizadas pelo sistema base.

4.1.4 OEBS Transparency Services

A camada OEBS Transparency Services concretiza o principal objectivo do projecto, caracterizando o Conector OutSystems para Oracle E-Business Suite, o qual pretende encapsular todas as interfaces com os sistemas Oracle, disponibilizando um conjunto de serviços de dados e acções aplicacionais às soluções a desenvolver, representadas pela camada superior, Composite Applications.

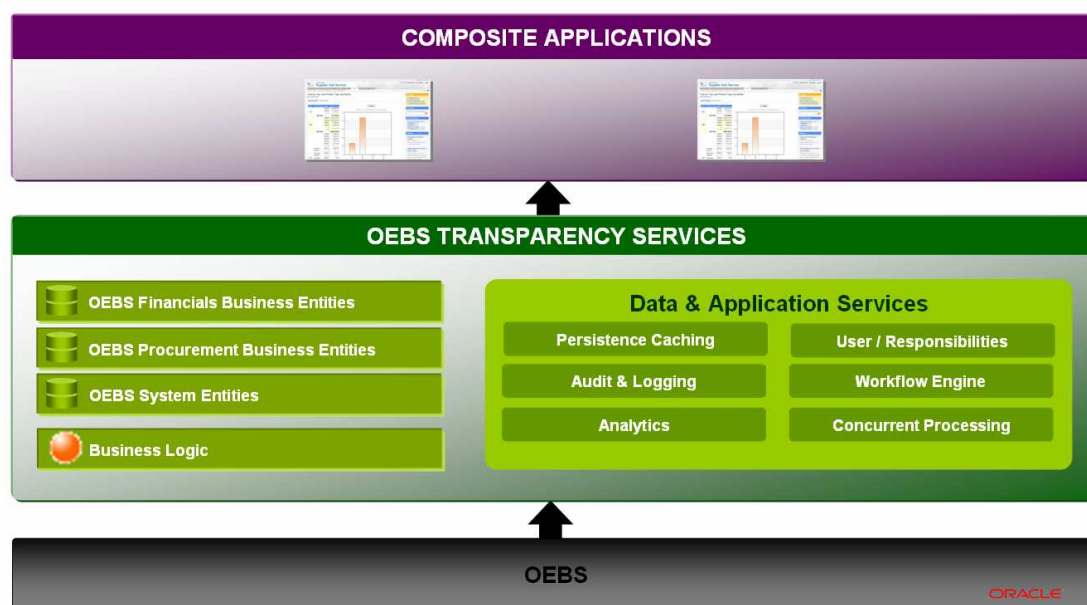


Figura 14 - Principais funções do OEBS Transparency Services

Esta camada inclui todas as entidades (tabelas) importadas de uma base de dados Oracle, que compõe a estrutura de armazenamento do OEBS, e igualmente o conjunto de todas as acções (métodos) criadas, quer por via automática (Generator), quer por via manual, permitindo o aproveitamento de toda a lógica de negócio constante no sistema base OEBS.

Os serviços que implementam as acções e funções do OEBS ficam disponíveis para o desenvolvimento de aplicações externas em torno dos sistemas já existentes, oferecendo abstracção ao nível da sua implementação interna e permitindo a sua integração e utilização de uma forma prática e simples.

4.1.5 Composite Applications

A camada Composite Applications engloba todas as aplicações que fazem uso da plataforma, explorando ou não as características da camada conectora. Esta camada contém toda a lógica de negócio que suporta o conjunto de características disponíveis para os utilizadores finais e encapsula as interfaces não *standard*.

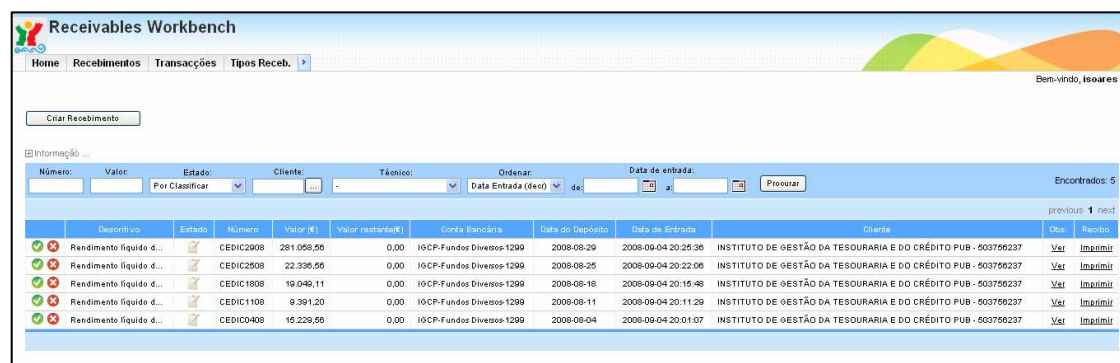


Figura 15 - Composite Application - Receivables Workbench

No âmbito do projecto foi desenvolvida a aplicação Receivables Workbench (Figura 15) que, tirando partido destas funções disponibilizadas pela plataforma, validou o modelo preconizado pelos vários componentes da arquitectura aqui apresentada.

4.2 Modelo de Segurança

O desenho inicial da arquitectura previa a utilização do acesso genérico à base de dados do OEBS (utilizador APPS), disponibilizado pelas aplicações em tempo de execução. Contudo, verificou-se que, se este utilizador fosse utilizado, qualquer aplicação desenvolvida na plataforma OutSystems teria os mesmos privilégios de execução e isso iria originar falhas de segurança. Este utilizador tem acesso vasto às tabelas e *APIs*, espelhando integralmente todo o conteúdo na camada conectora OEBS Transparency Services. Pelo facto de a plataforma poder ser partilhada por diferentes projectos/entidades de desenvolvimento, surge a necessidade de limitar o acesso aos recursos.

Por este motivo foi necessário incluir no desenho da arquitectura, um componente que permita garantir a segurança dos dados e o acesso apenas a utilizadores autorizados, ainda que em execução no mesmo contexto aplicacional.

Dado que a plataforma OutSystems só permite a definição de um único utilizador em tempo de execução, obriga a que este satisfaça as necessidades de qualquer aplicação, desenvolvida e disponibilizada pela plataforma. Definir um utilizador de acesso restrito a uma parte dos recursos é uma tarefa de difícil implementação uma vez que, por existirem dependências internas entre pacotes (*packages*), torna-se impraticável a determinação, de um modo simples e rápido, dos recursos a que devem ser concedidos acesso.

É pois necessário utilizar um acesso e dotá-lo das permissões e acessos correctos para cumprir todos os requisitos da plataforma e garantir ao mesmo tempo a necessária segurança e controlo de acesso aos dados das aplicações.

Este objectivo é dificultado pois estamos perante procedimentos ou funções da base de dados definidos em PL/SQL, que são executados com os direitos de quem os executa – *Authid Current_User* (e não de quem os criou – *Authid Definer*) [46]. A diferença entre estes dois modos consiste em:

- *Authid Definer* - independentemente do utilizador que efectua a chamada a um procedimento ou função, possuindo previamente privilégio de execução, este é executado com os direitos de quem o criou não sendo necessário garantir as dependências uma vez que foram verificadas aquando da sua criação/compilação;
- *Authid Current_User* - qualquer procedimento ou função é executado com os direitos do utilizador que o executa, sendo neste caso necessário garantir privilégios a todas as dependências existentes.

A pertinência desta questão reside no facto de, caso um utilizador efectue uma chamada a um procedimento com a definição de execução *Authid Current_User*, ser necessário a pré-atribuição de privilégios para execução de quaisquer dependências especificadas no corpo do procedimento, sob pena de ser desencadeada uma excepção de ausência de privilégios.

Para ultrapassar todas estas dificuldades, foi configurado o acesso da plataforma OutSystems à base de dados através de um utilizador com privilégios limitados e foi desenvolvido um componente responsável por criar uma camada de abstracção em PL/SQL ao nível da base de dados, servindo de interface entre as acções OutSystems geradas e as *APIs* oferecidas pelo Oracle E-Business Suite. Esta camada de abstracção é construída dinamicamente com a definição *Authid Definer*, sendo igualmente atribuídos, de forma automática, os privilégios necessários ao utilizador de execução definido na plataforma OutSystems.

Na figura 16 é apresentado um diagrama que ilustra o modelo de segurança desenhado.

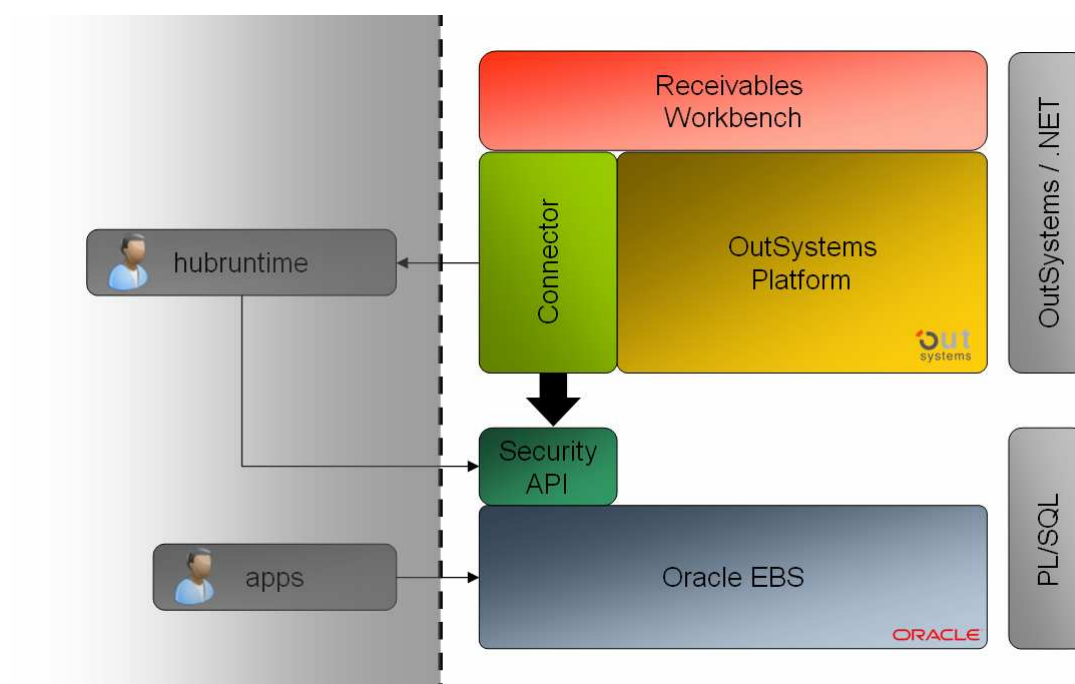


Figura 16 - Modelo de Segurança para limitar o acesso da plataforma OutSystems ao OEBS

Com este modelo, o utilizador de execução com privilégios limitados (*hubruntime*) tem acesso apenas à camada de interface de segurança *Security API*, suficiente para a interacção com as *APIs* do Oracle E-Business Suite, ocultando o respectivo código fonte e os respectivos dados. A questão levantada pela dependência de privilégios é assim ultrapassada pela aplicação na *API*, da característica *Authid Definer*, anteriormente descrita.

Capítulo 5

Trabalho Realizado e Resultados

Este capítulo relata o trabalho desenvolvido no projecto, enumerando os vários componentes concretizados e fazendo uma avaliação dos resultados obtidos.

5.1 Integração e Configuração

No período inicial deste projecto decorreu uma fase de autoformação, tendo incidido principalmente na aprendizagem da plataforma OutSystems e na obtenção da certificação em BASS (Building Applications in Service Studio) Nível 3. Incidiu igualmente na aprendizagem das ferramentas a utilizar no desenvolvimento do projecto e outras tecnologias periféricas necessárias.

Após este período inicial, seguiu-se a participação num projecto da PRÁXIA, numa organização da Administração Pública Portuguesa, de implementação e adaptação do Oracle E-Business Suite às necessidades da organização.

Os serviços prestados pela PRÁXIA incluíram duas grandes actividades:

- Tarefas de instalação e configuração inicial que incidiram na preparação da infraestrutura e na disponibilização de uma aplicação financeira aos utilizadores;
- Tarefas de apoio e manutenção evolutiva que contemplam além do apoio funcional e tecnológico, a extensão e desenvolvimento de novas funcionalidades.

A participação neste projecto incluiu a execução de tarefas de instalação e configuração inicial tendo sido integrado na equipa de desenvolvimento de aplicações periféricas em tecnologia OutSystems.

5.2 Connector e Aplicação Generator

Este percurso inicial permitiu a aquisição do conhecimento necessário, tanto a nível tecnológico como a nível do negócio, por forma a ter um olhar mais crítico e capaz sobre o objectivo principal do projecto.

O desenvolvimento do projecto ocorreu igualmente nas instalações de uma organização da Administração Pública, inserido numa equipa prestadora de serviços sobre Oracle E-Business Suite, o que permitiu a exploração e contextualização necessárias à concretização do mesmo.

O principal objectivo deste projecto passa pela criação de uma extensão à plataforma OutSystems para, de uma forma simples e sem codificação explícita, aceder à lógica de negócio instalada em bases de dados Oracle e, em particular, no OEBS.

As organizações que possuem aplicações instaladas sobre estas bases de dados, têm um investimento significativo realizado em lógica de negócio implementada junto aos dados, em procedimentos e funções realizadas em PL/SQL. Tipicamente, os procedimentos e funções PL/SQL implementam funções nucleares ao negócio, que estão bem testadas e operativas.

Como a plataforma OutSystems não providencia meios para invocar directamente procedimentos armazenados em bases de dados Oracle, sempre que se pretende estender as funções nucleares do sistema para novas soluções, é necessário desenvolver interfaces para a invocação das funções e procedimentos armazenados na base de dados. O desenvolvimento destas interfaces é um trabalho moroso e susceptível à inserção de erros, consumindo tempo dos programadores e não acrescentando valor às soluções desenvolvidas.

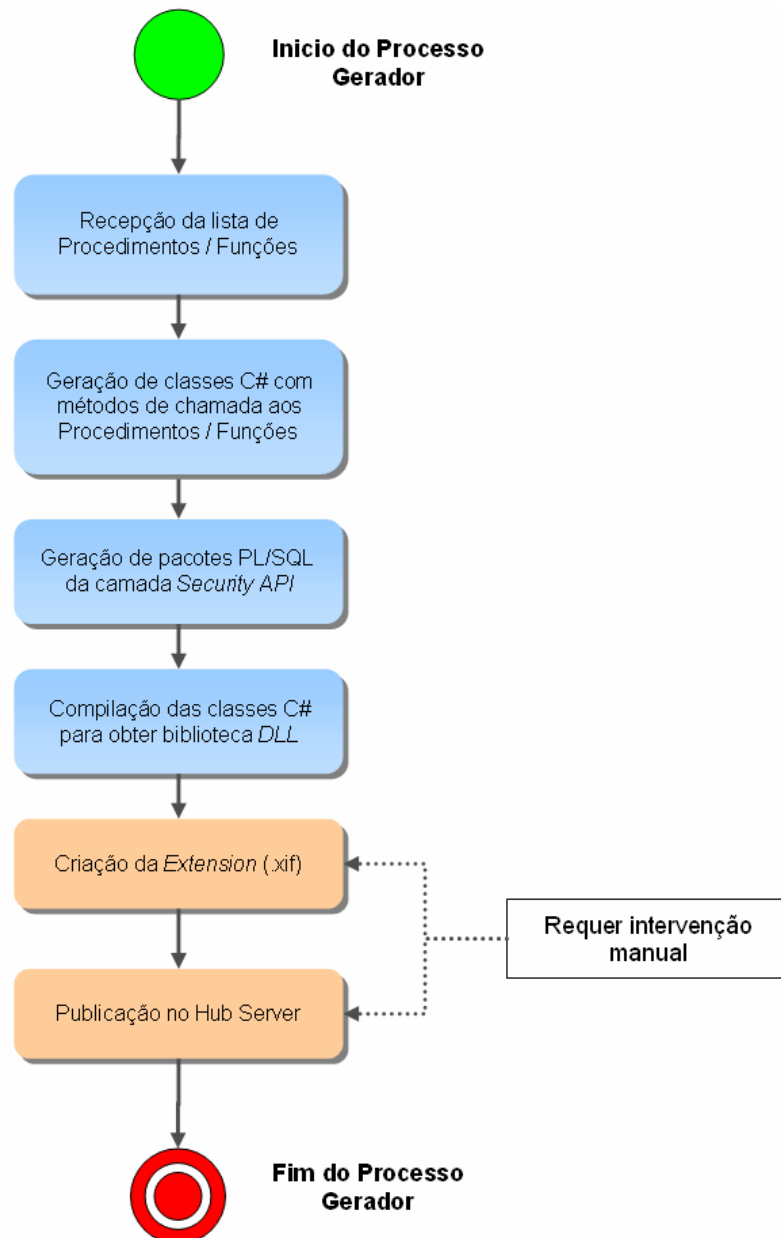
O desenvolvimento do Connector e aplicação Generator tenta ultrapassar estas questões, providenciando de forma simples, a geração dos pontos de conexão para as funções e procedimentos da base de dados Oracle. Cada ponto de conexão consiste numa chamada à base de dados relativa a um procedimento/função PL/SQL com os seus respectivos parâmetros. Estes possibilitam a utilização da lógica de negócio do OEBS directamente na plataforma de desenvolvimento OutSystems, libertando os programadores para aquilo que realmente interessa ao negócio: a implementação de funcionalidades que acrescentem valor à solução.

Para a geração dos pontos de conexão relevantes, foi necessário um estudo prévio da estrutura aplicacional do OEBS e do seu modelo de funcionamento. Este esforço teve como resultado a identificação de padrões estruturais de tabelas e procedimentos constantes nos pacotes de interface (*API Packages*) necessários ao desenvolvimento de aplicações periféricas ou extensão de aplicações já existentes.

Após análise das *APIs* e introspecção na base de dados, obtiveram-se valores na ordem dos 4 mil procedimentos/funções, referentes aos módulos financeiros do OEBS. As *APIs*, relativas a estes módulos, são identificadas através dos *packages* com prefixos *CE* (*Cash Management*), *GL* (*General Ledger*), *AP* (*Accounts Payable*), *AR* (*Accounts Receivable*) e *FA* (*Fixed Assets*).

Com base nesta informação, e por não ser uma tarefa comportável a selecção e criação manual destes pontos de conexão, foi planeada a automatização da sua geração, com a criação da aplicação Generator. Ao invés de se codificar um ponto de conexão, manualmente, para cada procedimento/função obtendo desta forma um conector forçosamente incompleto devido ao número e extensão das *APIs*, partiu-se para o desenvolvimento de um gerador capaz de criar estes pontos de conexão de forma automática, expondo-os à plataforma OutSystems. Ao conjunto de pontos de conexão, gerados e integrados na plataforma de desenvolvimento OutSystems por cada invocação do Generator, obtemos um Connector.

De seguida na figura 17 é apresentado um fluxo que ilustra o processo de geração implementado, seguindo-se o detalhe de cada passo do processo de geração.

**Figura 17 - Processo Gerador**

O mecanismo de geração construído consiste num conjunto de métodos desenvolvidos através da linguagem C# .NET, que recebem como parâmetro uma estrutura bem definida com a informação dos pacotes e procedimentos/funções PL/SQL a serem gerados. Com base nessa informação, são geradas classes C# (ficheiro .cs) individuais, correspondentes a cada pacote da base de dados, incluindo todos os métodos pretendidos. São simultaneamente gerados e compilados na base de dados os respectivos pacotes PL/SQL que compõem a camada de segurança e de interface (*Security API*), referida no Capítulo 4.

Após geração das classes C#, é necessária a compilação das mesmas obtendo-se assim uma biblioteca *DLL* com todos os métodos. Para disponibilizar à plataforma OutSystems todos estes pontos de conexão gerados, é necessária a criação de uma *Extension* (ficheiro .xif) e publicação através do ambiente de desenvolvimento OutSystems. Neste, são importados todos os métodos, ficando de imediato disponíveis no Hub Server, podendo, posteriormente, ser consumidos ao nível do Service Studio sob a forma de acções que implementam funções.

A adopção da linguagem C# .NET, para o desenvolvimento do conjunto de métodos que compõem o mecanismo gerador, baseou-se no facto de esta ser a forma mais rápida e eficaz para a obtenção de resultados práticos, na fase inicial de desenvolvimento. De salientar o facto de, todo o trabalho desenvolvido nesta vertente de *Extension* (.xif) requerer integração posterior na plataforma OutSystems e ter uma correspondência possível no Service Studio, evitando qualquer tipo de codificação manual. Desta forma, foi possível explorar esta potencialidade da plataforma, sendo notória a sua utilidade em situações em que, ao nível do código, sejam disponibilizadas bibliotecas cujos métodos ainda não tenham representação no Service Studio, e desta forma não possam ser usadas agilmente.

A aplicação Generator permite ainda uma pesquisa e selecção por pacotes, procedimentos e funções constantes na base de dados, respeitando as restrições impostas por problemas levantados na gestão dos tipos de dados dos parâmetros, garantindo desta forma uma total correspondência e funcionalidade. Esta aplicação, desenvolvida em tecnologia OutSystems, oferece uma interface intuitiva e de fácil utilização, simplificando o processo de selecção, compilação e criação do Connector. Uma vez criado e integrado, poder-se-á, ao nível do Service Studio, adicionar as acções que se pretendem para utilização nos fluxos das aplicações.

A aplicação Generator permite-nos obter os conectores que desejarmos, com os métodos que pretendermos, possibilitando a separação e organização de todas as acções por módulos ou pacotes. Desta forma, é igualmente possível disponibilizar apenas os métodos necessários a cada *eSpace*, evitando sobre exposição e garantindo assim segurança adicional.

As imagens seguintes ilustram um exemplo de execução da aplicação Generator, desde o momento de pesquisa até à geração do código e sua disponibilização.

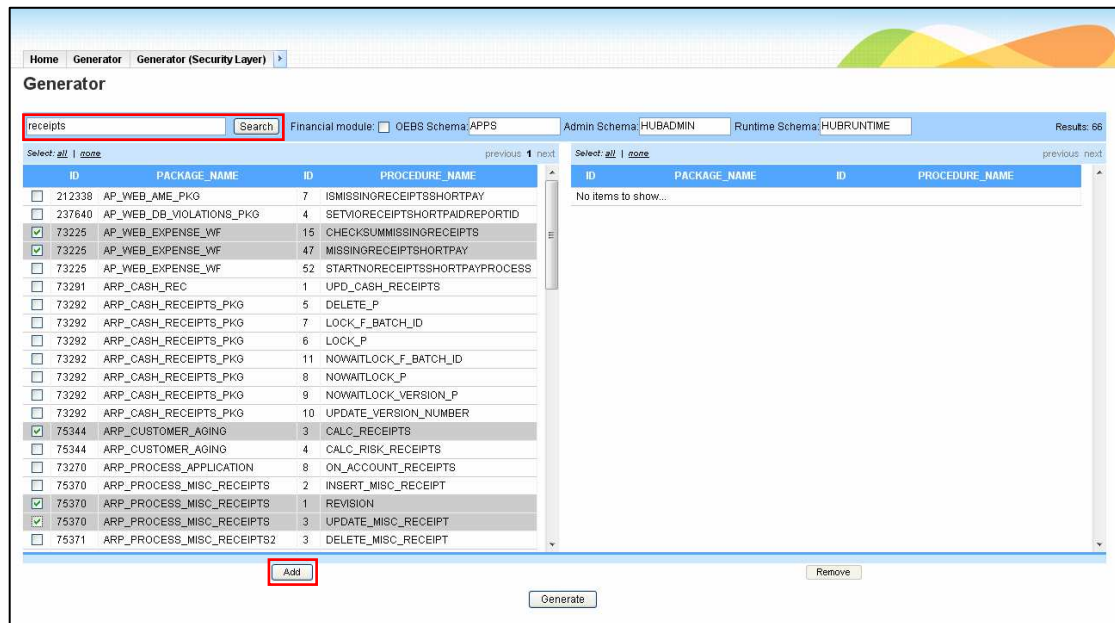


Figura 18 - Generator - Pesquisar e Adicionar

Na figura 18 é apresentada a interface da aplicação Generator onde consta a funcionalidade de pesquisa e selecção de procedimentos ou funções para serem adicionados à lista de métodos a serem gerados.

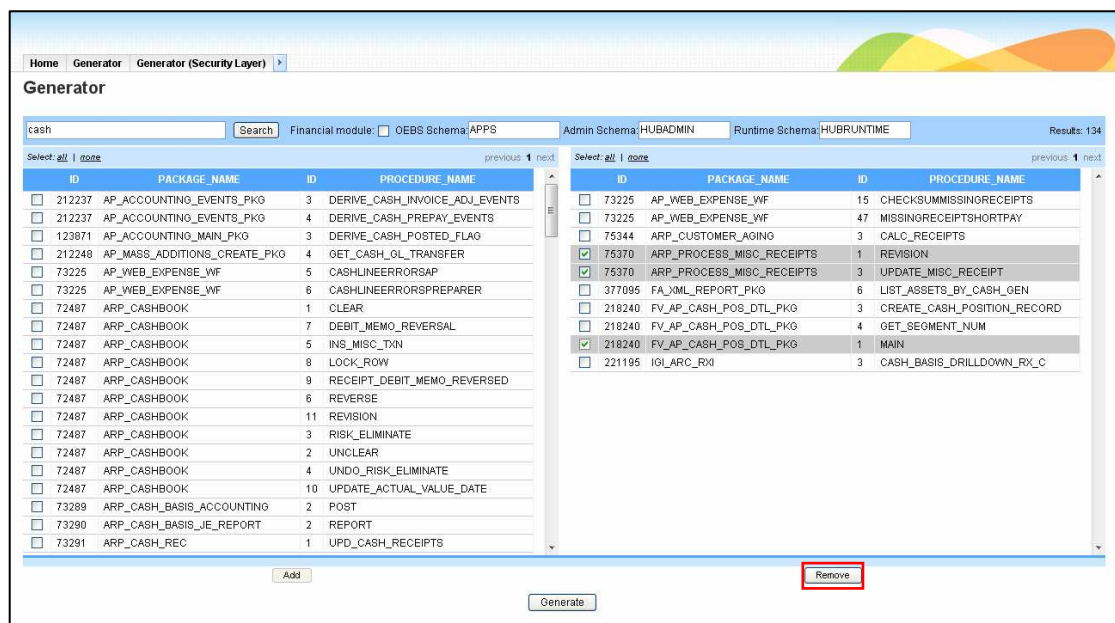


Figura 19 - Generator - Remover

Na figura 19, após nova pesquisa e adição, verificamos a funcionalidade de remoção da lista de métodos a serem gerados.

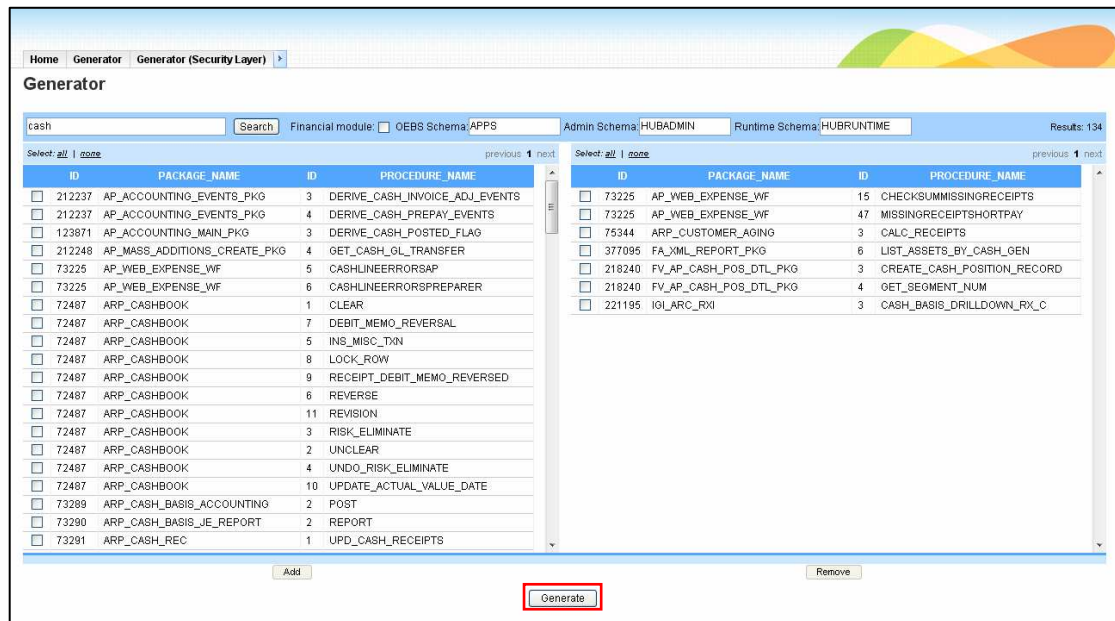


Figura 20 - Generator - Gerar

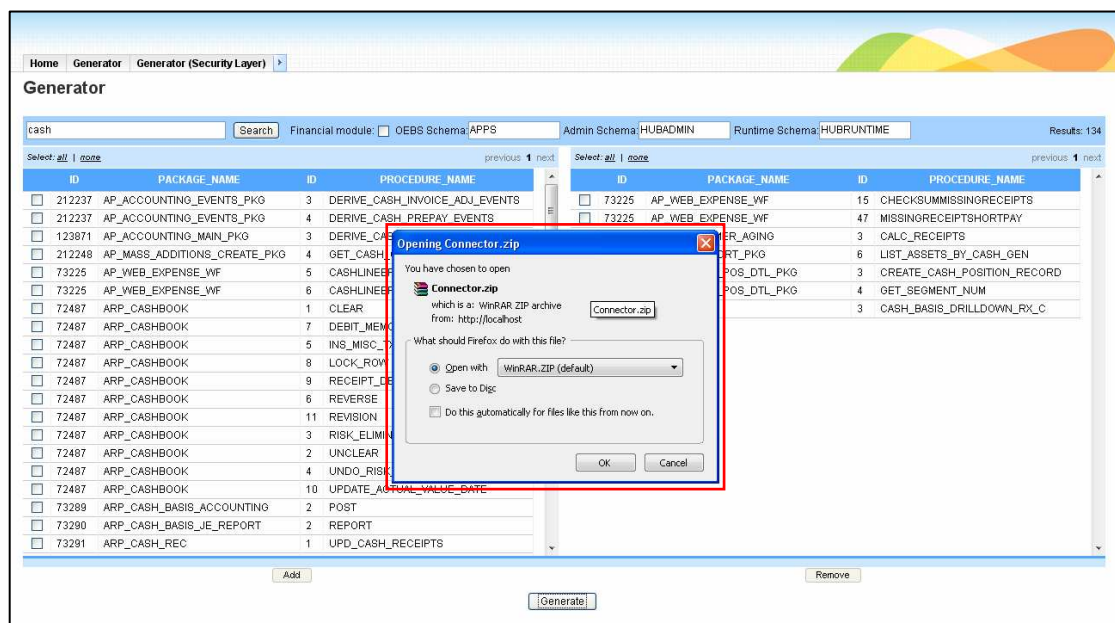


Figura 21 - Generator - Download

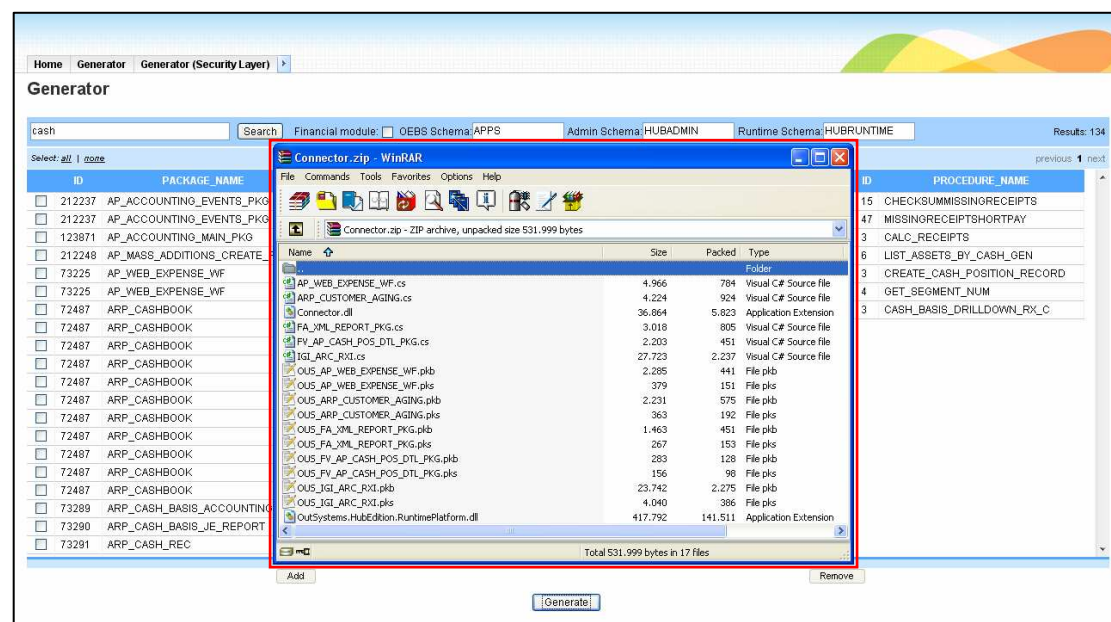


Figura 22 - Generator - Ficheiro gerados

Nas figuras 20, 21 e 22 é mostrada a acção que desencadeia a geração e compilação do código relativo aos procedimentos e funções seleccionados, após finalizado o processo de pesquisa e selecção. É retornado o produto do processo Generator, que corresponde ao conjunto de todos os ficheiros gerados e compilados pela aplicação.

A interface Generator permite ainda pesquisar apenas por procedimentos ou funções respeitantes ao módulo financeiro (*checkbox - Financial module*), acelerando assim o processo de selecção e criação de um Connector para o Oracle Financials.

Para a construção da camada de segurança (*Security API*) e geração dos métodos C# que compõem o Connector, são necessários os dados dos utilizadores/esquemas (OEBS, Admin e Runtime Schema) em uso pela plataforma OutSystems para acesso à base de dados Oracle.

Na construção do Connector e Generator surgiram múltiplos obstáculos que condicionaram o seu desenvolvimento. Segue-se uma descrição dos mais relevantes acompanhada das acções realizadas para colmatar os problemas encontrados.

Tipos de dados

Um dos problemas surgidos durante a fase de desenvolvimento do Connector foi a compatibilidade dos tipos de dados entre as tecnologias OEBS e OutSystems/.NET.

Uma vez que as *APIs* do OEBS manipulam múltiplos tipos de dados, muitos deles nativos e manipulados apenas num contexto interno, foi efectuado um estudo sobre todos os tipos de dados afim de determinar o seu mapeamento possível com a tecnologia em que iria assentar a plataforma OutSystems, neste caso a .NET Framework.

A tabela que se segue ilustra a correspondência entre os tipos de dados usados pelas *APIs* do OEBS, e os tipos de dados oferecidos pela tecnologia .NET Framework:

ORACLE	.NET
BFILE	Byte[]
BINARY_DOUBLE	
BINARY_FLOAT	
BINARY_INTEGER	
BLOB	Byte[]
CHAR	String
CLOB	String
DATE	DateTime
FLOAT	Decimal
INTERVAL DAY TO SECOND	TimeSpan
INTERVAL YEAR TO MONTH	Int32
LONG	String
LONG RAW	Byte[]
MLSLABEL	
NCHAR	String
NCLOB	String
NUMBER	Decimal
NVARCHAR2	String
OBJECT	
PL/SQL BOOLEAN	Bool
PL/SQL RECORD	
PL/SQL TABLE	
RAW	Byte[]
REF	
REF CURSOR	
ROWID	String
TABLE	
TIME	
TIME WITH TIME ZONE	
TIMESTAMP	DateTime
TIMESTAMP WITH LOCAL TIME ZONE	DateTime

ORACLE	.NET
TIMESTAMP WITH TIME ZONE	DateTime
UROWID	
VARCHAR2	String
VARRAY	

Tabela 2 - Tipos de Dados

Embora só seja possível, devido a este condicionamento, a manipulação de parte das *APIs* do OEBS, a utilização dos tipos de dados manipuláveis (com correspondência na tabela 2) garante uma total correspondência e funcionalidade na chamada aos procedimentos e funções das *APIs*.

Valores nulos

Outro problema encontrado foi o facto de a plataforma OutSystems não contemplar o valor nulo – ‘*null*’ para qualquer tipo de dados, ao contrário do que se verifica na base de dados Oracle. Ao invés disto a plataforma OutSystems assume um valor por omissão para qualquer tipo de dados de um atributo de uma tabela ou parâmetro de um método.

Num sistema OEBS, em contraste com a tecnologia OutSystems e a linguagem C# .NET, para além de ser possível a representação do valor ‘*null*’, os procedimentos e funções também podem incluir parâmetros opcionais (definidos com a cláusula “*default* [valor]”). O facto de não ser possível representar o valor ‘*null*’ ao nível do ambiente de desenvolvimento Service Studio, impossibilita também, ao nível do código C# .NET, determinar se um parâmetro recepcionado com valor idêntico ao atribuído por omissão pela plataforma OutSystems, deverá ou não ser considerado nulo. Este comportamento leva a que não se consigam accionar os valores por omissão dos parâmetros opcionais na base de dados.

Para contornar esta característica da plataforma OutSystems, no processo de geração das classes C#, é adicionado e associado um parâmetro adicional, do tipo *boolean*, a cada parâmetro opcional relativo a um procedimento/função da base de dados.

Desta forma é possível ao nível do Service Studio, indicar explicitamente quais os parâmetros opcionais e respectivos valores, que deverão ser incluídos na chamada à

base de dados, conseguindo desta forma que os restantes parâmetros opcionais assumam o seu valor por defeito.

Mapeamento de entidades OEBS

Outro problema identificado na plataforma OutSystems corresponde ao mapeamento dos nomes das entidades OEBS importadas através do Integration Studio.

O mecanismo automático de importação de entidades não efectua um mapeamento directo. Ao invés disso este, por razões internas, não mantém os nomes originais dos atributos das entidades, sacrificando os dois últimos caracteres quando o nome do atributo é superior a 28 caracteres.

Este comportamento impossibilita o uso de funcionalidades práticas e úteis, oferecidas pelo ambiente de desenvolvimento, uma vez que, quando usadas, resultam em erros de execução pela correspondência inexistente entre a representação gerada pela plataforma OutSystems e as entidades OEBS.

Para contornar este problema é re-implementado o mapeamento recorrendo a funcionalidades avançadas, algo que em condições normais não é necessário.

Importação de métodos

Outro aspecto relevante ainda a referir, prende-se com o facto de o ambiente de desenvolvimento Integration Studio, ao importar métodos através de uma biblioteca *DLL* para a geração de acções e posterior utilização no Service Studio, assumir todos os parâmetros dos métodos como obrigatórios. Este comportamento torna-se problemático quando estamos perante chamadas a procedimentos/funções da base de dados com dezenas de parâmetros, em que muitos deles são opcionais, não sendo reflectida essa especificação no Service Studio.

Desta forma é necessária uma intervenção manual para redefinir cada um dos parâmetros opcionais.

5.3 Aplicação Receivables Workbench

Em resposta a uma solicitação do organismo da Administração Pública - Turismo de Portugal, para criação de um *software* para automação e simplificação de procedimentos relacionados com a gestão de recebimentos, foi apresentada uma proposta para a criação de uma aplicação a ser desenvolvida na plataforma OutSystems sobre funcionalidades disponibilizadas pelo Oracle Financials.

Especificamente no Turismo de Portugal, existe um grande investimento anterior em funções e procedimentos em PL/SQL, quer sejam os procedimentos *standard* instalados pelo Oracle E-Business Suite, quer sejam procedimentos e funções desenvolvidos à medida do cliente para outras aplicações periféricas sobre os dados do Oracle E-Business Suite.

A reutilização destes componentes em novas aplicações a desenvolver sobre o Oracle Financials é condição essencial para a realização destas novas aplicações.

A aplicação Generator anteriormente concebida, toma um papel preponderante expondo todas as funcionalidades do OEBS necessárias para integrar as aplicações a desenvolver em OutSystems.

Neste sentido foi desenvolvida a aplicação Receivables Workbench. Uma aplicação cujo objectivo é o de simplificar a forma como os utilizadores interagem com o OEBS, eliminando trabalhos duplicados ou redundantes e reduzindo o tempo necessário para:

- Registrar recebimentos;
- Criar transacções de receita;
- Classificar transacções de receita;
- Aplicar recebimentos a transacções de receita;
- Emitir recibos.

Através de uma simples interface *Web* é possível realizar estas operações, simplificando drasticamente a operação do Oracle E-Business Suite e permitindo a utilizadores não treinados a realização de operações de tesouraria.

A simplificação de processos proporcionada por esta aplicação permite reduzir os custos de operação, reduzindo o tempo necessário para tratar e classificar documentos, libertando os utilizadores de tarefas mecânicas para tarefas com maior criação de valor.

Para o armazenamento da informação gerada pela Receivables Workbench durante a sua execução, foi desenhado um modelo de dados próprio que permite manter toda a informação organizada e disponível para correcção, consulta e processamento.

Esta estrutura de armazenamento serve igualmente de interface, entre as interfaces do utilizador e as tabelas definitivas do Oracle E-Business Suite, na medida em que só após processamento da informação, recorrendo tanto à lógica desenvolvida ao nível do Service Studio, como à lógica em PL/SQL do Oracle Financials, é que todos os dados se tornam parte integrante do OEBS, ficando acessíveis a outros módulos do sistema.

A figura 23 apresenta o modelo de dados do Receivables Workbench:

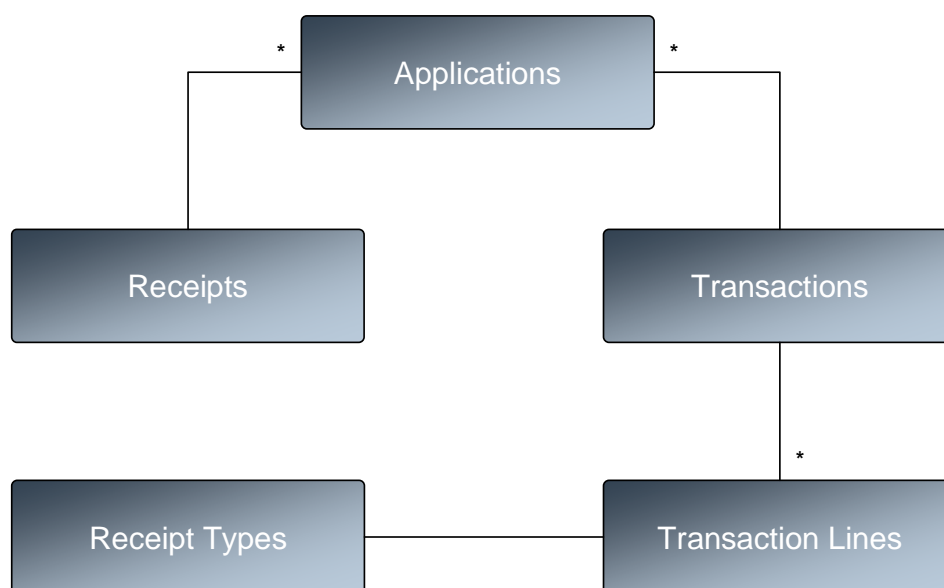


Figura 23 - Receivables Workbench - Modelo de Dados

Nesta figura podemos verificar a organização e relação entre as entidades que compõem a aplicação e armazenam a informação relativa aos recebimentos (*Receipts*), transacções (*Transactions*) e tipos de recebimento (*Receipt Types*).

Em traços gerais, o registo de um recebimento representa um valor de um cheque, transferência bancária ou numerário proveniente de um cliente, o qual se encontra desta forma a liquidar uma dívida. Associado a cada recebimento poderão estar mais do que uma transacção com vista a cobrir o valor total do recebimento. Cada transacção terá tantas linhas de transacção quantas forem necessárias por forma a discriminarem totalmente o valor da transacção. Cada linha de transacção inclui uma classificação contabilística que permite a catalogação e identificação da proveniência da receita em questão.

A tabela de tipos de recebimento, contém combinações de classificações contabilísticas, permitindo automatizar, simplificar e acelerar a classificação das linhas de transacção de um recebimento.

Após classificação completa e válida, uma transacção fica apta para processamento assíncrono através de serviços disponibilizados pelo sistema OEBS e encapsulados pelo Connector.

De seguida são apresentados os ecrãs principais da aplicação:

The screenshot displays the 'Receivables Workbench' application. At the top, there are navigation tabs: 'Home', 'Recebimentos', 'Transacções', and 'Tipos Receb.'. Below the tabs is a 'Criar Recebimento' button. A search bar labeled 'Informação...' contains fields for 'Número', 'Valor', 'Estado', 'Cliente', 'Técnico', 'Ordenar', and 'Data de entrada'. A 'Pesquisar' button is located to the right of these fields. Below the search bar, a table lists receivables with columns: 'Descrição', 'Estado', 'Número', 'Valor (€)', 'Valor restante(€)', 'Conta Bancária', 'Data do Depósito', 'Data de Entrada', 'Cliente', 'Obs.', and 'Recibo'. The table shows four entries, all with the state 'Por Classificar'.

Descrição	Estado	Número	Valor (€)	Valor restante(€)	Conta Bancária	Data do Depósito	Data de Entrada	Cliente	Obs.	Recibo
Rendimento líquido d...	Por Classificar	CEDIC2908	281.056,56	0,00	IGCP-Fundos Diversos 1299	2008-08-20	2008-09-04 20:25:36	INSTITUTO DE GESTÃO DA TESOUREARIA E DO CRÉDITO PUB - 503756237	Ver	Imprimir
Rendimento líquido d...	Por Classificar	CEDIC2908	22.336,56	0,00	IGCP-Fundos Diversos 1299	2008-08-25	2008-09-04 20:22:06	INSTITUTO DE GESTÃO DA TESOUREARIA E DO CRÉDITO PUB - 503756237	Ver	Imprimir
Rendimento líquido d...	Por Classificar	CEDIC1808	10.040,11	0,00	IGCP-Fundos Diversos 1299	2008-08-18	2008-09-04 20:15:48	INSTITUTO DE GESTÃO DA TESOUREARIA E DO CRÉDITO PUB - 503756237	Ver	Imprimir
Rendimento líquido d...	Por Classificar	CEDIC1108	9.391,20	0,00	IGCP-Fundos Diversos 1299	2008-08-11	2008-09-04 20:11:29	INSTITUTO DE GESTÃO DA TESOUREARIA E DO CRÉDITO PUB - 503756237	Ver	Imprimir
Rendimento líquido d...	Por Classificar	CEDIC0408	15.229,58	0,00	IGCP-Fundos Diversos 1299	2008-08-04	2008-09-04 20:01:07	INSTITUTO DE GESTÃO DA TESOUREARIA E DO CRÉDITO PUB - 503756237	Ver	Imprimir

Figura 24 - Receivables Workbench - Gestão de Recebimentos

Na figura 24 é apresentada a página de gestão de recebimentos. Podem ser efectuadas diferentes combinações de pesquisa, editar recebimentos e criar novos. Como podemos verificar, os recebimentos listados encontram-se todos no estado “Por Classificar”. Como exemplo de execução, a figura 25 apresenta o formulário de criação de um recebimento.

Figura 25 - Receivables Workbench - Criar Recebimento

No formulário da figura 25 são fornecidos dados tais como um identificador, valor total, identificação do cliente que está a efectuar o pagamento assim como uma classificação, opcional, com base em três tipos que pretendem, deterministicamente, identificar a origem do recebimento. Caso não seja efectuada a classificação nesta fase, o recebimento é criado, no entanto não é processada qualquer transacção associada ao mesmo.

Uma vez desencadeada a acção de criação, e caso a classificação contabilística se encontre completa, o recebimento segue para processamento automático. Caso não se encontre completa, é apresentado o recebimento em questão para finalização da classificação da transacção.

Figura 26 - Receivables Workbench - Classificar Transacção

Na figura 26, é apresentado o recebimento acabado de criar, as transacções que tem associadas (uma única neste caso) e as linhas de classificação contabilística da transacção. Cada linha de uma transacção incorpora um conjunto de sete segmentos contabilísticos (Organização, Produto, Centro de Responsabilidade, Acção, Conta, Mercado e Fonte de Financiamento), os quais permitirão classificar a receita. Através destes segmentos contabilísticos, será possível efectuar uma análise futura mais fina e rica.

Nesta figura verificamos um preenchimento da classificação contabilística incompleto (Item e Acção sem informação). Isto deve-se ao facto de a classificação escolhida, aquando da criação do recebimento, se encontrar definida desta forma, deixando a sua conclusão ao critério do operador.

Esta aplicação possui igualmente uma interface para gestão independente de transacções, permitindo o registo das mesmas sem a necessidade da existência de um recebimento. Desta forma, quando é criado um novo recebimento, poder-se-á aplicar uma transacção já existente no sistema, com valor ainda disponível por aplicar. Foi também criada uma interface para a gestão dos tipos de recebimento.

Na figura 27 é apresentada uma ilustração do ambiente de desenvolvimento oferecido pelo Service Studio com uma das acções da aplicação que, no seu fluxo, executa *queries* bem como outras acções disponibilizadas pelo Connector.

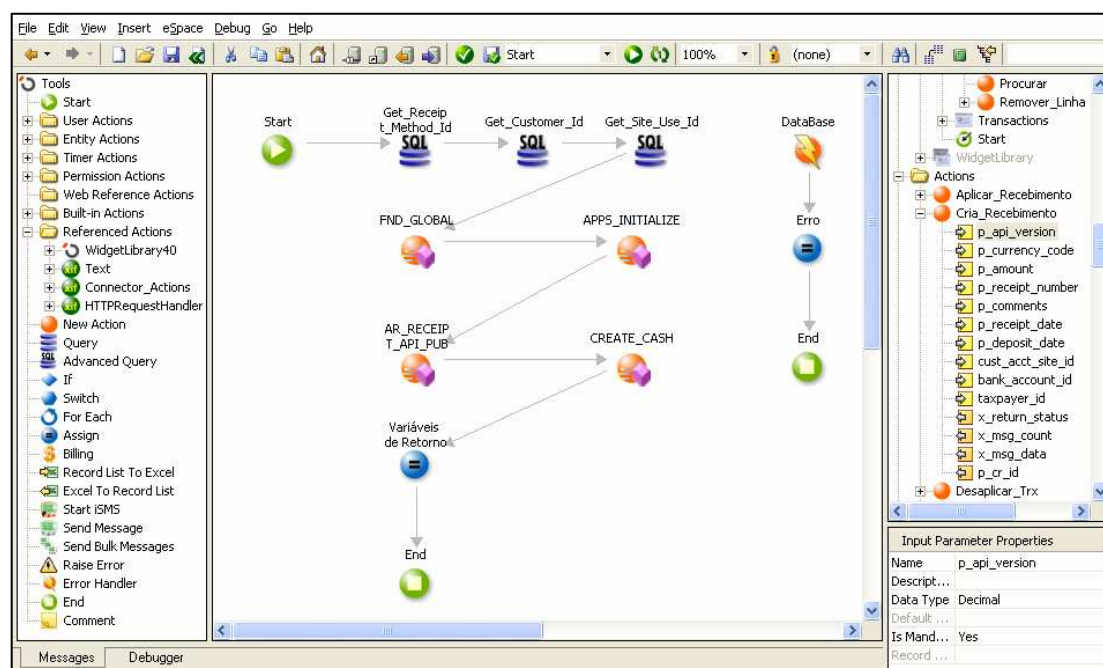


Figura 27 - Ambiente de desenvolvimento Service Studio - Fluxo da Action Criar_Recebimento

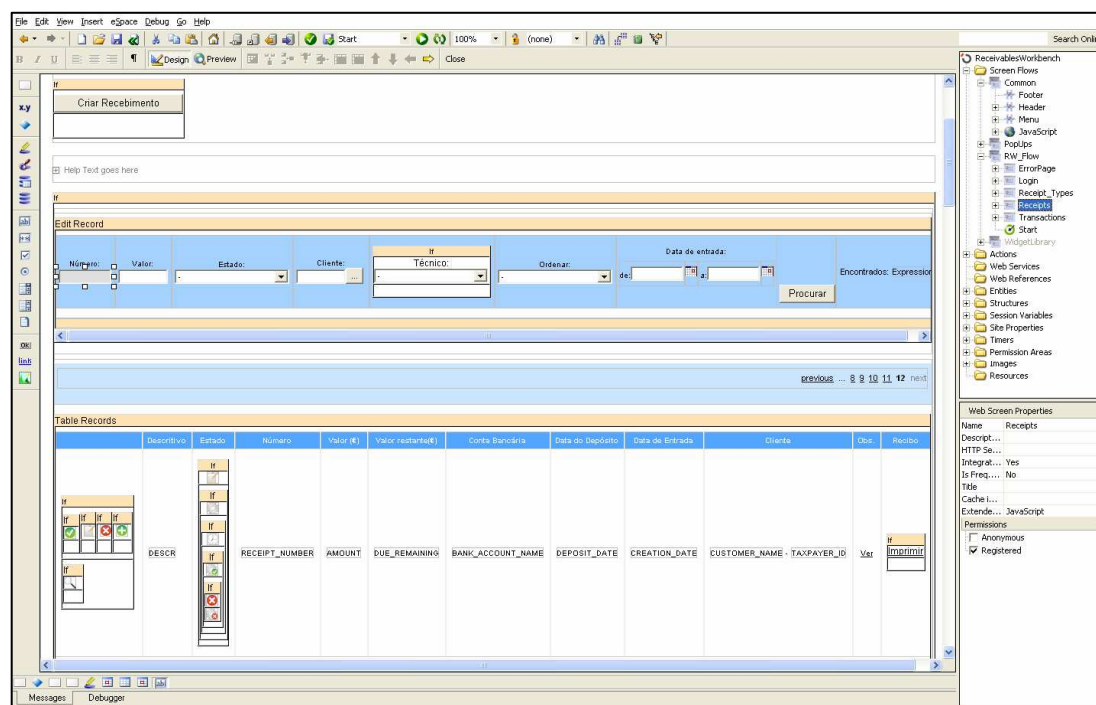


Figura 28 - Ambiente de desenvolvimento Service Studio - Interface do utilizador para a gestão de recebimentos

Na figura 28 é apresentada uma visão de parte da interface do utilizador relativa à página de gestão de recebimentos com a construção de uma tabela (*Table Records*) para listar os recebimentos, assim como a estrutura que permite pesquisar os recebimentos, aplicando filtros.

5.4 Resultados Alcançados

A concepção da aplicação Generator permitiu representar 68% da lógica de negócio constante nos sistemas Oracle E-Business Suite na plataforma OutSystems.

A agilidade e rapidez (em poucos minutos) associada à obtenção de um Connector, oferecendo abstracção da lógica de negócio ao nível da codificação, permite a qualquer programador aplicar mais eficazmente o seu tempo ao desenvolvimento de aplicações (componente lógica e interface do utilizador), focando-se no desenvolvimento de tarefas de maior criação de valor.

A abstracção associada à conectividade da plataforma OutSystems ao Oracle E-Business Suite contribui para uma amostragem e entrega de resultados mais célere, uma vez que o programador não necessita de se preocupar com este tipo de questões.

O investimento inicialmente empregue no desenvolvimento da aplicação Generator revela-se desta forma compensador, uma vez que foi possível desenvolver a aplicação Receivables Workbench num período de tempo inferior àquele que seria necessário despende, através de outra tecnologia ou precisando de construir os pontos de conexão manualmente.

A tabela 3 resume as estimativas obtidas para o tempo necessário para o desenvolvimento da aplicação Receivables Workbench com diferentes abordagens.

Tecnologia	Estimativa de desenvolvimento	Tempo de desenvolvimento
Nativa F&R / 4GL	10 semanas ^① 1,5 FTE	
OutSystems	7 semanas ^② 1,5 FTE	
OutSystems + Connector	6 semanas 1,5 FTE	7 semanas 1,2 FTE

① - Estimativa obtida pelo TP

② - Estimativa obtida da equipa OutSystems da PRÁXIA
(Delivery Manager) e ferramentas de estimação da plataforma

FTE - Full-Time Equivalent

Tabela 3 - Receivables Workbench - Análise do tempo estimado e efectivo de desenvolvimento

Pelos valores recolhidos verificámos que, efectivamente, há neste caso, um ganho potencial com a utilização do Connector no desenvolvimento da aplicação Receivables Workbench.

A experiência e informação transmitida pelos colaboradores da PRÁXIA relativa a aplicações desenvolvidas no passado, em diferentes tecnologias, e em uso não só no

Turismo de Portugal como também em outras instituições, vieram confirmar os benefícios de utilização e desenvolvimento através da tecnologia OutSystems.

A aplicação Receivables Workbench, actualmente em utilização no Turismo de Portugal, revelou um nível de aceitação elevado, confirmando ser uma aplicação apelativa, intuitiva, prática e eficaz, facilitando o processo administrativo associado. A implementação desta aplicação resultou numa quase total ausência de resistência por parte dos operadores que se encontravam familiarizados com procedimentos diferentes dos agora adoptados, abraçando com facilidade a mudança.

Para além da aplicação Receivables Workbench, foi também desenvolvida uma outra aplicação que integra *Web Services* de um sistema externo ao OEBS implementado no Turismo de Portugal. A tecnologia OutSystems revelou-se mais uma vez ágil tornando o processo de desenvolvimento célere.

Capítulo 6

Conclusão e Trabalho Futuro

O principal objectivo deste projecto foi o desenvolvimento de um conector de integração do sistema Oracle E-Business Suite com a plataforma OutSystems, com vista a facilitar o desenvolvimento de futuras aplicações *Web*, através da tecnologia OutSystems, para integrarem o sistema aplicacional Oracle E-Business Suite.

O Connector obtido permite criar aplicações periféricas aos sistemas OEBS, podendo estas explorar toda a estrutura de base de dados que centraliza a informação gerada por estes sistemas. O Connector possibilita igualmente explorar a lógica de negócio utilizada pelas aplicações Oracle E-Business Suite, reduzir a complexidade e minimizar as necessidades específicas de integração das aplicações periféricas.

Devido à extensão das interfaces (*APIs*) que representam a lógica de negócio do OEBS, foi desenvolvida a aplicação Generator, responsável por gerar de uma forma automática os pontos de conexão que constituem o Connector, sem que para isso seja necessário qualquer tipo de codificação.

Um dos aspectos que condicionou o desenvolvimento do projecto, traduzindo-se na aplicação de tempo adicional, decorreu do facto de a plataforma OutSystems não ser integralmente compatível com as representações de tipos de dados e parâmetros de procedimentos/funções do OEBS. Para tal foi estabelecida uma correspondência entre os tipos de dados compatíveis garantindo assim uma total operacionalidade entre estas duas tecnologias.

Outro aspecto que condicionou o desenvolvimento do projecto foi o facto de a plataforma OutSystems não contemplar o valor nulo – ‘*null*’ para qualquer tipo de dados, impossibilitando a transmissão de parâmetros com valor nulo a um sistema OEBS. Para contornar esta questão foi implementada uma solução que passa pela criação de parâmetros extra que permitem, ao nível do ambiente de desenvolvimento, indicar explicitamente se um parâmetro deverá ou não assumir valor nulo.

Estas situações foram sendo ultrapassadas no decorrer do projecto, pelo que o objectivo do projecto se demonstra alcançado, não só pelas capacidades da aplicação Generator e do Connector, como também por estes poderem ser aplicados futuramente em qualquer organização que tenha implementado um sistema OEBS e uma plataforma OutSystems. Com o Connector é possível construir aplicações que explorem a informação gerada pelo OEBS assim como a lógica de negócio existente, reduzindo a complexidade e minimizando as necessidades específicas de integração destas aplicações periféricas, de forma ágil e célere, permitindo assim alcançar os requisitos de rapidez e custo reduzido impostos pelas leis do mercado.

Este projecto, realizado no âmbito da disciplina Projecto em Engenharia Informática do Mestrado em Engenharia Informática, revelou ser uma experiência enriquecedora tanto a nível académico como a nível profissional, permitindo conhecer e aprofundar o conhecimento das tecnologias envolvidas, contribuindo para a aquisição do *knowhow* imprescindível para a entrada e estabelecimento no mercado de trabalho.

Trabalho Futuro

Embora os objectivos tenham sido alcançados com resultados satisfatórios, algumas melhorias e extensões poderão ser realizadas no futuro.

Foi solicitada à OutSystems, a análise de uma possível integração da aplicação Generator com a ferramenta Integration Studio por forma a automatizar por completo o processo de geração do Connector e respectiva publicação no Hub Server da plataforma. Uma das soluções apresentadas para a geração da *Extension* (.xif) com base na biblioteca *DLL*, gerada actualmente pela aplicação Generator, é a disponibilização de uma linha de comandos para esse efeito. Actualmente esta melhoria encontra-se em fase de desenvolvimento.

Outra questão em análise, e no seguimento da anterior, é a necessidade de actualização do Connector após correcção ou adição de métodos à biblioteca já existente. Esta questão não é de trivial resolução uma vez que, para ser mantida a integridade do Connector, todas as referências de acções que não sejam modificadas terão que ser mantidas inalteradas sob pena de invalidar a utilização das acções em uso nos fluxos dos *eSpaces*. Esta é mais uma questão que passa directamente pela intervenção e acesso privilegiado aos mecanismos de geração e manutenção de extensões através do Integration Studio.

Como trabalho futuro, pretende-se ainda aperfeiçoar a aplicação Receivables Workbench transformando-a numa aplicação multi-organização. Para tal será necessário trabalhar alguns aspectos, entre eles, a adaptação às diferentes definições possíveis de segmentação contabilística usada pelas organizações.

É ainda plano da PRÁXIA tornar tanto a aplicação Generator como o Connector num produto comercializável. Para isso será imprescindível ultrapassar as questões anteriormente identificadas, com o intuito de tornar o trabalho até agora realizado num produto robusto e de qualidade.

Bibliografia

- [1] Agile Solutions for SAP: <http://www.outsystems.com/sap/>
- [2] BRAUDE, E. J., Software Engineering, An object-Oriented Perspective. John Wiley & Sons, Inc. 2001.
- [3] Business Intelligence:

LUHN, H.P., “*Business Intelligence System*”
<http://www.research.ibm.com/journal/rd/024/ibmrd0204H.pdf>, IBM Journal, 1958.
- [4] CMP Technology: <http://www.cmp.com/>
- [5] Codies Awards: <http://www.siiia.net/codies/2008/>
- [6] CRM:

WAILGUM, T., “*ABC: An Introduction to CRM*”,
http://www.cio.com/article/40295/ABC_An_Introduction_to_CRM/, CIO, 2007.
- [7] ERP:

WAILGUM, T., “*ABC: An Introduction to ERP*”,
http://www.cio.com/article/40323/ABC_An_Introduction_to_ERP/, CIO, 2007.
- [8] Java Application Server: http://en.wikipedia.org/wiki/Application_server
- [9] Jolt Awards Finalists: <http://www.joltawards.com/finalists.html>
- [10] Metodologia de Desenvolvimento em Espiral:

Disciplina de Projecto de Sistemas de Informação. Guião das aulas teóricas.
Docentes FCUL/DI/SI. 2007.

-
- [11] Microsoft Dynamics: <http://www.microsoft.com/dynamics/default.mspx>
- [12] Microsoft Internet Information Server:
http://en.wikipedia.org/wiki/Internet_Information_Services
- [13] Microsoft Navision: <http://www.microsoft.com/dynamics/nav/default.mspx>
- [14] Microsoft SQL Server: <http://www.microsoft.com/sql/default.mspx>
- [15] Microsoft: <http://www.microsoft.com>
- [16] Oracle Application Development Framework (ADF):
<http://www.oracle.com/technology/products/adf/index.html>
- [17] Oracle Database: <http://www.oracle.com/database/>
- [18] Oracle Developer Suite (iDS):
<http://www.oracle.com/technology/products/ids/index.html>
- [19] Oracle Discoverer:
<http://www.oracle.com/technology/products/discoverer/index.html>
- [20] Oracle E-Business Suite: <http://www.oracle.com/applications/e-business-suite.html>
- [21] Oracle Forms: <http://www.oracle.com/technology/products/forms/index.html>
- [22] Oracle Internet Application Server (iAS):
<http://www.oracle.com/appserver/index.html>
- [23] Oracle JDeveloper: <http://www.oracle.com/technology/products/jdev/index.html>
- [24] Oracle Reports: <http://www.oracle.com/technology/products/reports/index.html>
- [25] Oracle: <http://www.oracle.com/>
- [26] OutSystems (Downloads):
OutSystems Express Edition surpasses 10.000 downloads in 2007,
http://www.outsystems.com/agile/contentdetail.aspx?contentname=news_10000downloads&folderpath=%5croot%5ccontents%5ccorporate%5clandingpages%5cnews
[ews](#) , San Ramon - CA, 14 January 2008.

[27] OutSystems (Jolt Awards):

OutSystems Recognized By Prestigious SIIA Codie and Jolt Awards,

<http://www.outsystems.com/agile/Content.aspx?ContentName=SIIACodieAwards&FolderPath=%5cRoot%5cContents%5cCorporate%5cLandingPages%5cNews>,

San Ramon - CA, 15 February 2008.

[28] OutSystems (Magic Quadrant):

OutSystems positioned in Integrated Service Environment Magic Quadrant,

http://www.outsystems.com/agile/contentdetail.aspx?contentname=news_gartner&folderpath=%5croot%5ccontents%5ccorporate%5clandingpages%5cnews,

Redwood City - CA, 06 February 2006.

[29] OutSystems Supplier Self Service:

<http://www.outsystems.com/agile/Solution.aspx?FolderPath=%5CRoot%5CContents%5CCorporate%5CBusinessSolutions%5CSupplierSelfService>

[30] OutSystems:

<http://www.outsystems.com/>

http://www.outsystems.com/CMS_BackOffice/ResourceDownload.aspx?ResourceName=OutSystemsPlatformOverviewDatasheet

[31] PHC Enterprise: <http://www.phc.pt/portal/programs/ewpview.aspx?codigo=terp>

[32] PHC Software: <http://www.phc.pt/>

[33] PLM:

GOULD, L., “*The ABCs of PLM*”,

<http://www.autofieldguide.com/articles/120310.html>, AutoFieldGuide, 2003.

GOULD, L. S., “*Additional ABCs About PLM*”,

<http://www.autofieldguide.com/articles/120506.html>, AutoFieldGuide, 2005.

[34] Primavera ERP:

<http://www.primaverabss.com/corporate/PortalRender.aspx?PageID={323aeb11-84fd-4b72-9ee9-049fb4b76ae9}>

[35] Primavera: <http://www.primaverabss.com/>

[36] SAP Business Suite: <http://www.sap.com/solutions/business-suite/>

[37] SAP Enterprise SOA: <http://www14.sap.com/about/press/factsheets/esoa.epx>

[38] SAP ERP líder nacional:

SANTOS, A. “A importância dos ERP para o mercado”,
http://www.ver.pt/conteudos/Detalhes_Clipping_Sector.aspx?Ev=1784, VER
(Valor, Ética e Responsabilidade), 28 Março 2008.

[39] SAP líder mundial:

”SAP lidera mercado de software empresarial no mundo”,
http://press.blogs.com/presshitportuguese/2006/08/sap_lidera_merc.html,
PressHit.com, Business Intelligence, Press Releases, SAP | Permalink. 28 Agosto
2006.

[40] SAP: <http://www.sap.com/>

[41] SCHWABER, Ken, “*Agile Project Management with Scrum*”, Microsoft Press,
Washington, 2004.

[42] SCM:

WORTHEN, B., “*ABC: An Introduction to Supply Chain Management*”,
http://www.cio.com/article/40940/ABC_An_Introduction_to_SCM/, CIO, 2007.

[43] Scrum:

[http://www.controlchaos.com/about/?SID=8ef7eb5b2a069a2710abef27d02c851f
&SID=7da824062baf60b8e78ec5f99836f092](http://www.controlchaos.com/about/?SID=8ef7eb5b2a069a2710abef27d02c851f&SID=7da824062baf60b8e78ec5f99836f092)

[44] Software & Information Industry Association: <http://www.siiia.net/>

[45] SRM:

“*Supplier Relationship Management*”
http://en.wikipedia.org/wiki/Supplier_relationship_management, Wikipedia.

[46] “*User's Guide and Reference - PL/SQL*”, 10g Release 1 (10.1), Oracle, 2003.

[47] Web Services: <http://www.w3.org/2002/ws/>, W3C.